

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: N 2612 – Elektrotechnika a informatika

Studijní obor: 3902T005 - Automatické řízení a inženýrská informatika

Řídící aplikace pro kontrolu kvality výrobků

Management applications for products quality inspection

Diplomová práce

Autor: Bc. Hynek Moravec

Vedoucí práce: Doc. Ing. Ivan Jaksch, CSc.

Konzultant: Ing. Jiří Horčíčka,
Ing. Otto Havle, CSc. MBA

V Liberci dne: 2. 1. 2011

Originál zadání DP

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č.121/2000 o právu autorském, zejména §60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé BP a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení, apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licence k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum:

Podpis:

Poděkování

Na tomto místě bych rád poděkoval všem, kteří mi pomáhali při tvorbě mé diplomové práce. Především rodině, která mi poskytla klidné zázemí a morální a materiální podporu po dobu celého studia. Dále pak jistě slečně, které mě velice podpořila při tvorbě programu a psaní DP. A na závěr panu Ing. Otto Havle, CSc. MBA za možnost podílet se na vývoji tohoto projektu a celému kolektivu firmy FCC průmyslové systémy a.s. za podporu.

Abstrakt

V češtině

Název práce: Řídící aplikace pro kontrolu kvality výrobků

Autor: Bc. Hynek Moravec

Katedra (ústav): Ústav řízení systémů a spolehlivosti

Vedoucí práce: doc. Ing. Ivan Jaksch, CSc.

Abstrakt: Cílem práce je vyvinout aplikaci pro řízení zařízení, jenž kontroluje kvalitu výrobků. První část seznamuje s jednotlivými prvky zařízení a s jejich programováním. Druhá část popisuje vytvořenou aplikaci, její ovládání a přístupy k datům.

Klíčová slova: programování, C#, průmyslové zařízení, robot, kamera

V angličtině

Title: Management applications for products quality inspection

Author: Bc. Hynek Moravec

Department: Institute of systems control and reliability management

Supervisor: doc. Ing. Ivan Jaksch, CSc.

Abstract: The goal of the thesis is to develop controlling application for a device, which checks the final products quality. First section acquaints with (introduces) the individual parts of the machinery (equipment) and way of their programming. Second section describes the application, controlling system and the access to the data.

Keywords: programming, C#, industrial installation, robot, camera

Obsah

Prohlášení.....	3
Poděkování.....	4
Abstrakt.....	5
1. Úvod.....	8
2. Seznámení se zařízením.....	10
3. Robot.....	12
3.1. První metoda řízení	16
3.2. Druhá metoda řízení robota.....	18
3.3. Programování robota.....	19
4. Kamera.....	22
4.1. Popis.....	22
4.2. Použitá kamera	24
4.3. Programování	26
4.4. OCR.....	27
5. Adamy.....	28
5.1. Řízení Adamů.....	31
6. Druhy měření	33
7. Uživatelská aplikace	37
7.1. Stupně přidělených práv.....	38
7.1.1. Přihlašovací obrazovka	39
7.1.2. Supervizor.....	40
7.2. Výběr projektu a dílu	42
7.3. Vložení informací.....	43
7.4. Náповěda.....	44
7.5. Inspekce dílu	46
7.6. Kontrola.....	48

7.7.	Identifikace.....	50
7.8.	Měření	51
7.9.	Výstupní protokol.....	52
8.	XML.....	53
8.1.	Ukázka.....	54
8.2.	Struktura XML v aplikaci	56
8.2.1.	XML pro výběr projektu.....	56
8.2.2.	XML pro daný díl	58
8.2.3.	Výstupní XML.....	62
9.	Závěr	63
	Seznam použité literatury	65
	Příloha A - Hash	66
	Příloha B – výstupní protokol.....	67
	Příloha C – obsah vloženého CD	68

1. Úvod

Vyvíjené zařízení, které bude na následujících stránkách popsáno, umožňuje zvýšit kvalitu a konkurenční schopnost dané firmy, jelikož bude možno efektivně kontrolovat kvalitu jednotlivých sérií daných výrobků. V dnešním konkurenčním boji mezi jednotlivými výrobci je kladen vysoký důraz na kvalitu a bezchybnost dodávaného zboží, proto se každá firma snaží zaručit správnost a minimalizovat kazivost svého výstupního produktu. Po době kladení vysokého důrazu na výslednou cenu, kdy se zaváděli nové a rychlejší technologie, mnohdy na úkor kvality, přišli odběratelé (automobilky, kompletace elektroniky, atd.) s požadavky na dodržování kvalitativních norem a zároveň s nutností podepsání tzv. Quality Agreementu, kde jsou stanoveny sankce a povinnosti v případě nedodržení kvality (odběr špatných dílů, třídění výrobků, případně podpora při opravě v místě odběratele). Odběratelé též určují tzv. PPM (parts per milion) určující maximální množství vadných dílů na jeden milion dodaných. Automobilový průmysl tlačí dodavatele na snížení tohoto čísla téměř k hodnotě 0. V případě překročení této hodnoty je dodavatel nucen zaplatit nemalé pokuty, proto jsou výrobci nuceni investovat do zařízení, jenž jim umožní efektivnější a spolehlivější kontrolu vyrobených dílů.

Jako výstupní člen kontroly je většinou používán zaměstnanec, jenž se spoléhá na svůj úsudek a zkušenosti při vizuálních kontrolách a na správnost použití měřidla při rozměrových měřeních. Člověk však může udělat chybu vlivem únavy, nepozornosti, či z důvodu zachování jistého výrobního standartu linky. Nejen z těchto důvodů jsou na výstupní kontroly instalovány automatické měřicí jednotky, které zaručí, že daný díl bude vždy se stejnou kvalitou zkontrolován a změřen. Pokud je možné toto zařízení používat i pro více výrobků, sníží se tak i náklady na instalaci různých zařízení pro různé díly.

Zařízení nesnižuje míru vad, které se vyskytnou během výrobního procesu, ale má za úkol snížit míru vadných sérií, které se dostanou až na výstup z továrny a tím i k odběrateli, který tak ztrácí důvěru k tomuto výrobcí. Toto snížení se děje pomocí opakovatelnosti měření se stále stejnou přesností v řádkách setin milimetrů a vizuální kontrolou s téměř bezchybnou jistotou odhalení špatného prvku.

Jelikož zařízení není umístěno přímo ve výrobním procesu, ale je odděleno, je nutné zajistit přísun potřebných dílů do měřicího prostoru. Toto dodání dílu obstarává pověřená osoba (operátorem), která vkládá díly, jenž je potřeba zkontrolovat, podle požadavků daného výrobního procesu. Je tak nutné zajistit, aby se díly vždy přesně vložily na předem definované místo a aby byl vybrán v řídicí aplikaci správný díl pro kontrolu, jelikož dílů bude několik a budou mít rozdílné tvary. Díly se vkládají do výměnných roštů mimo zařízení, je tak usnadněna manipulace a je možno si díly do roštu předpřipravit v době, kdy probíhá měření na jiném dílu.

Takto vyvinuté zařízení by mělo mít přijatelnou dobu návratnosti financí, zejména za ušetření placení pokut při nesplnění požadavků na kvalitu, nebo také za výdaje na zaměstnance provádějící kontrolu. Zařízení by mělo být schopné plně zastoupit ruční provádění kontrol a především značně urychlit výstupní kontrolu na jeden díl. Tím by mohlo být zkontrolováno větší množství dílu a případně snížena hodnota PPM.

Na následujících stránkách budou rozepsány jednotlivé důležité části zařízení, jejich stručný popis, charakteristika, důvod zvolení a nástin jejich programování. Dále bude popsáno chování uživatelské aplikace, která je stěžejním tématem této diplomové práce. V poslední části bude ukázána struktura dat v XML souborech, jejich práce s nimi a důvod použití tohoto formátu pro ukládání dat. Závěr této práce se věnuje možnostem vylepšení aplikace, její osvědčení v provozu a novým požadavkům ze stran zadavatele.

2. Seznámení se zařízením

Aby bylo zařízení univerzální pro kontrolu různých dílů, je nutné zajistit, aby snímací zařízení, zde kamera, bylo pohyblivé a dokázalo zaujmout ideální pozici pro pořízení snímku. Proto bude kamera namontována na koncovém rameni robota, jenž dokáže zajistit, aby se kamera dostala pokud možno do co největšího počtu míst, která se budou na daném výrobku kontrolovat. Pro správné detekování rozměrů a objektů ve snímku je nutné zařídit vhodné osvětlení, proto bude v zařízení umístěno několik nezávislých světel, která budou ovládána podle požadavků na pořízení snímku z aplikace. Celý systém zařízení se bude ovládat z jednoho centrálního počítače, který bude zároveň sloužit i k ovládání řídicí aplikace.

Celé zařízení se bude skládat z několika dílčích prvků, jenž lze rozdělit na část technickou a část počítačovou. Tyto části spolu úzce souvisí, avšak popsány budou především jen části počítačové, které budou přímo ovlivňovat vyvíjenou řídicí aplikaci. Všechny tyto počítačové části spolu budou komunikovat po počítačové síti ethernet, což bude umožňovat snadnou modulaci zařízení během jejího vývoje, ale i dohled nad jednotlivými prvky v zařízení, jejich monitoring, revizi či údržbu i z jiných míst.

Jednotlivé počítačové prvky budou nezávisle vyvíjeny tak, aby se urychlil vývoj celého zařízení, a v samotném závěru budou postupně zkompleťovány, aby utvořily jeden funkční prvek. Vývoj zařízení bude upravován podle aktuálních požadavků, které se budou postupně formulovat v závislosti na zjištění nových poznatků ze stran jak zadavatele, tak i ze stran vývoje. Aplikace pro řídicí systém bude psána v jazyce C# v prostředí Visual Studio. Tato aplikace bude spravovat veškeré jednotlivé části systému a bude zajišťovat i jejich vzájemnou komunikaci.

Veškerá data pro aplikaci budou načítána ze souboru ve formátu XML, především data o jednotlivých dílech jako jejich popis, hmotnost či požadované rozměry. Při výběru daného dílu, jenž provádí operátor, se poté načte potřebné XML a zobrazí požadované informace. Aplikace bude umožňovat jednoduchou editaci daného XML, především úpravu informací vztahujících se k danému dílu.

Velice vhodnou funkcí je možnost kalibrace jednotlivých konstant pro převod mezi pixely, jenž budou poskytovány při měření z kamery, a milimetry, které jsou výslednou hodnotou pro měření. Tato kalibrace by se měla provádět automaticky, tj.

zapsáním přesných hodnot do tabulky a během kalibrování daného dílu, výpočet převodní konstanty z pixelu na milimetry.

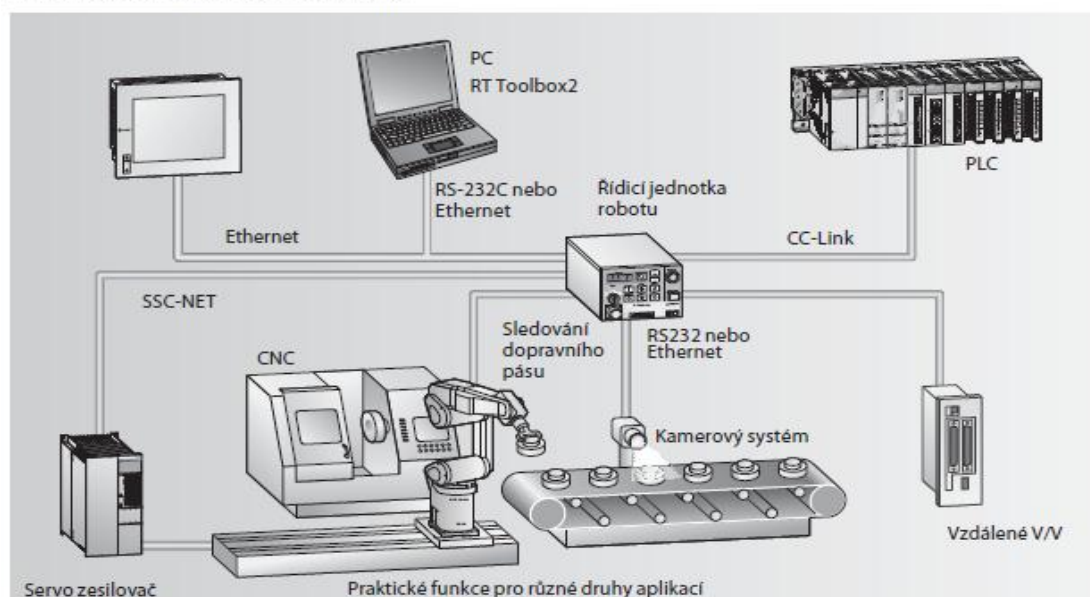
Po ukončení měřicího procesu, by měla aplikace umožnit tisk výstupního protokolu, který bude obsahovat údaje o osobě provádějící měření, informace o měřeném dílu, data zadaná operátorem (váha dílu, číslo) a především změřené hodnoty s případným vizuálním odlišením hodnot, které jsou mimo toleranci.

3. Robot

Dle zadavatele byl vybrán robot od japonské společnosti Mitsubishi Electric Corporation, vyznačující se inovativním přístupem, prvenstvím ve vývoji elektrických zařízení pro informační a komunikační technologie, vesmírný a satelitní výzkum, energetiku, spotřebního elektroniku, atd.

Roboti japonské firmy se vyznačují vysokým výkonem a produktivitou orientovanou na potřeby zákazníka. Obsahují pouze jedno připojovací místo pro napájení a ovládání pohonů, což urychluje jejich nasazení v provozu. Příruby pro montáž koncového chapadla jsou v souladu s normou ISO 9409-1 a umožňují tak připojení chapadel od nejrůznějších výrobců. Každý robot je dodáván s řídicí jednotkou MELFA, která umožňuje řízení dodatečných os (pojezdy), a je možno tyto jednotky připojit do strukturované datové sítě (CC-Link či ethernet). Roboti od Mitsubishi se vyznačují mnoha technologickými vlastnostmi umožňujícími přesnější, spolehlivější a ekonomičtější chod. Nejvýznamnějšími je automatická optimalizace zrychlování a brždění, kontinuální posuv či kompenzace gravitace.

Příklad uspořádání robotického systému

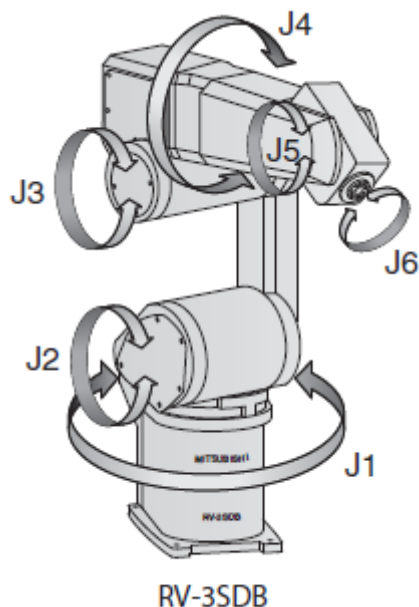


Obrázek 3-1: Robotické systémy Melfa

Možné uspořádání robotického systému s automatickým řízením pomocí kamerového systému (pokud kamera detekuje objekt, robot jej odebere) je znázorněno na obrázku 3-1. Řídící jednotka ovládá PLC zařízení, které řídí výrobní procesy. Vstupy a výstupy ovládané pomocí řídicí jednotky robota mohou sledovat pohyb dopravního pásu. Veškeré tyto činnosti jsou předávány po ethernetu do zobrazovací jednotky či do počítače. Tato sestava velice urychlí výrobní proces a ušetří výdaje, jelikož celý systém je centralizován do řídicí jednotky robota. Ta umožňuje i paralelní řízení více procesů, tudíž není žádná činnost opožděna či pozastavena při vykonávání činnosti jiné.

Stupně volnosti označují hlavní parametr robota - definují, jak velice se dokáže robot přizpůsobit a dostat se do všech míst daného pracovního prostoru. Jeden stupeň volnosti je jeden možný druh nezávislého pohybu v daném místě. Např. loket člověka má jeden stupeň volnosti, který nám umožňuje pohyb sklápění předloktí.

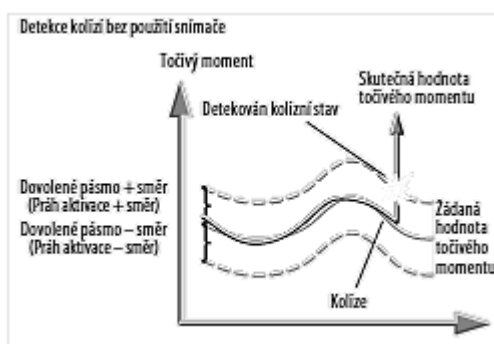
Pohyb můžeme rozdělit na posuvný (translační) či rotační (sférický). Pro manipulaci v malém a předem definovaném prostoru je nejlépe zvolit sférického robota s 5, s případně 6 stupni volnosti, které nám umožní se dostat do jakéhokoliv místa. Robot se 6 stupni volnosti s rotačními osami je zobrazen na obrázku 3-2.



Obrázek 3-2: Stupně volnosti

Bezpečnost robota vůči okolnímu prostředí, zejména vůči člověku, je jednou z nejdůležitějších priorit při jeho návrhu a umístění do výrobního procesu. Je nutné pokud možno zamezit přístup člověka do pracovního prostoru robota. Pokud se člověk musí pohybovat v prostoru robota, je potřeba, aby bylo možné robota bezpečně zastavit a nejlépe, aby bylo umožněno odpojení servopohonů od napájení. Pokud však již dojde ke středu s překážkou, je možné nastavit citlivost na kontakt a robota okamžitě zastavit. Toto bezpečnostní opatření se provádí nastavením rozsahu proudové zátěže pro jednotlivé osy (serva) a při překročení této hranice se robot automaticky zastaví. Bezpečnostní zastavení probíhá zastavením pohybu serv pomocí brzdy umístěné na společné ose, tím je docíleno velice rychlého a bezpečného zastavení a tím minimalizování poškození či úrazy. Řídící jednotka je též vybavena bezpečnostními vstupy od externích čidel. Externí čidla mohou být především signál od nouzového tlačítka umístěného poblíž robota, či snímač otevření bezpečnostních dveří či klece. Oba tyto signály musí okamžitě zastavit serva a uvést robota do klidového stavu. Je nutné si uvědomit, že robot musí být po celou dobu zastavování napájen, aby fungovaly mechanické brzdy a až po té je vhodné odpojit jeho napájení.

Na obrázku 3-3 je znázorněn průběh proudové zátěže při pohybu servopohonu. Vodorovná osa znázorňuje pohyb (úhel natočení či translační pohyb), svislá osa znázorňuje aktuální proudový odběr. Čárkované osy značí dovolenou toleranci proudové zátěže, jelikož odběr proudu se při pohybu může měnit v závislosti na objektu v chapadle, či změně těžiště robota (vysunutí os apod.). Pokud však odběr proudu prudce vzroste nad tuto povolenou mez, je zaznamenána kolize, jelikož servopohon se pokouší zvýšit svojí výkonnost a tím i se zvyšuje i proud potřebný k napájení daného serva.



Obrázek 3-3 : detekce kolize

Souřadnice se do robota mohou zadávat ve dvou formátech. Joy – který řídí úhly natočení pro jednotlivé osy. XYZ – jsou souřadnice v prostoru pro koncové rameno.

Robot není řízen přímo pomocí aplikace, ale jeho pohyby jsou koordinovány přes řídicí jednotku robota, která upravuje rychlost a směr pohybu jednotlivých os. Takto uspořádané řízení nám umožňuje pouze zaslat patřičná data do řídicí jednotky a ta již sama určí, jakým způsobem se robot dostane do daného místa. Nemůžeme však přesně určit, po jaké dráze se robot bude dostávat do daných míst, je proto nutné určovat tzv. mezipolohy, které upraví jeho požadovanou trajektorii, aby nedošlo ke kolizi robota s kontrolovaným výrobkem či se stěnou celého zařízení.

Robot by nebyl pouze manipulátorem pro kameru, ale také by prováděl testy výlisku a to pomocí trnu, který by byl na něm nainstalován. Tímto trnem bude zajišťováno do děr a kontrolováno, zda nejsou zaslepené z důvodu špatného odlití výlisku. Trn bude umístěn v pružném pouzdře se snímačem, který bude reagovat na pohyb trnu v důsledku kontaktu s místem, které mělo být průchozí. Detailnějšímu popisu se věnuji v kapitole Druhy měření.

3.1. První metoda řízení

Pro potřeby aplikace byla prvotně zvolena metoda řízení robota pomocí speciálních procedur uložených přímo v řídicí jednotce včetně jednotlivých trajektorií a přesných bodů, do kterých se má robot dostat. Aplikace by poté pouze odesílala robotovi žádost o „další“ pozici a ten by po najetí do ní pouze oznámil, že jí dosáhl. Aplikace by poté prováděla další potřebné operace (pořízení snímku, vyhodnocení dat). Tímto způsobem by bylo zabezpečeno korektní ovládání robota, jelikož procedura by byla vytvořena přímo v řídicí jednotce robota. Tak by byly i informace o aktuální poloze ihned zpracovány a řízení by probíhalo bez nebezpečí ztráty dat např. z důvodu vypadnutí či zničení některého ze síťových prvků.

Pro případ měření děr byl na digitální vstup robota přiveden signál od tohoto snímače, který určoval, že je snímač sepnut. V momentě sepnutí by musel být robot okamžitě zastaven, aby nedošlo k poničení trnu či samotného výlisku. Bylo tak zareagováno na zalitou díru a musela by se tato informace sdělit řídicí aplikaci. Byl proto vytvořen komunikační protokol, kdy robot bude oznamovat pomocí binární hodnoty True či False že dojel na danou pozici. V případě, že měření je prováděno pomocí kamery, oznámí robot hodnotu True v momentě, kdy dojel na danou pozici, a mohou být tak prováděny další operace. V případě měření díry by byla odeslána hodnota True, pokud by trn dojel bez záznamu snímače do spodní části díry (díra je proto v pořádku), hodnota False by znamenala, že trn nedojel na požadované místo a díra je tudíž zalita.

Bylo nutné zajistit, aby se robot dokázal z kteréhokoliv místa bezpečně vrátit do domácí pozice, tato situace mohla nastat, když operátor ukončil měření korektním způsobem z programu. Muselo být zajištěno, aby robot neustále věděl, ve které pozici se zrovna nachází (stačila i koncová měřicí pozice) a jak se z této pozice dostat do pozice domácí bez kolize. To by mohlo nastat, pokud by byla díra pro měřicí trn ve vodorovné rovině a tudíž by do ní bylo i vodorovně zajížděno. V tento moment nejde poslat robota do domácí pozice, která je od této umístěna v horizontálním směru, robot by zničil jak trn, tak i samotný výlisek. Bylo nutné nejprve z díry bezpečně vysunout trn a poté poslat robota do domácí pozice. Riziko kolize ovšem mohlo nastat i v momentě, kdy je robot natočen kamerou poblíž výlisku a jeho okamžitý nájezd do domácí polohy by znamenal opět kolizi z důvodu postupného otáčení všech os. Bylo nutné určit

robotovi bezpečné pozice, pro nekolizní vzdálení se od výlisku a až v těchto pozicích dát robotovi příkaz pro návrat do domácí pozice.

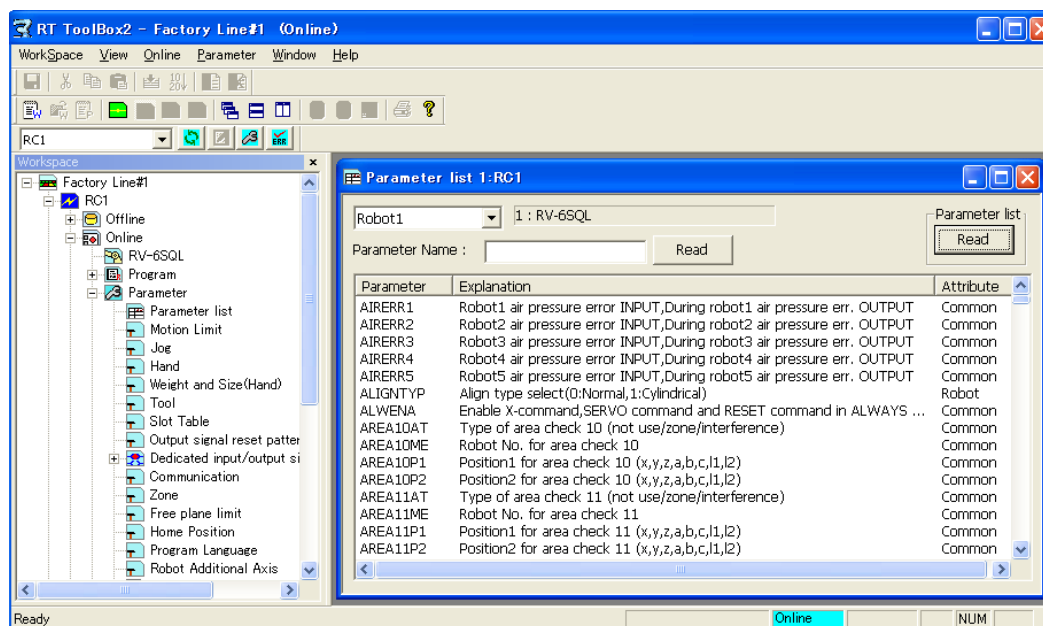
Všechny tyto pozice by byly napevno zadány v jednotlivých procedurách v řídicí jednotce robota. Před samotným měřením by se pouze aktivoval daný program v robotovi a pro nájezd na pozice by se používaly pouze příkazy od aplikace ve formátu „Další“ (nájezd na další měřící pozici), či „stop“ (pro návrat robota na domácí polohu). Tímto způsobem by odpadlo veškeré řízení robota v aplikaci a samotná aplikace by se velice zjednodušila. Velkou nevýhodou bylo, že chyběla zpětná vazba od robota s informací, ve kterém místě se momentálně nachází. Aplikace se tak nemohla dozvědět, zda je robot již na pozici č. 8, či číslo 9. Informace byla pouze taková, že je robot na pozici. Tuto nevýhodu by bylo možno ošetřit tak, že by robot odesílal s informací o dosažení pozice i informaci o číslu pozice. Aplikace by tak měla jistotu, že je robot opravdu na požadovaném místě. Další značnou nevýhodou bylo komplikované upravování souřadnic jednotlivých pozic pro robota. Nešlo tyto pozice zadávat mimo robota, např. do textového souboru a ten jen vložit do řídicí jednotky. Postupem vývoje přišly požadavky, aby se dráha robota dala modifikovat za běhu aplikace, např. vynechávání měřících míst, především test děr, upravování rychlosti robota, přesnost dojezdů apod. Tyto požadavky si vynutily zásadní úpravu řízení robota a to řízení z aplikace, která bude zasílat dané požadavky o cílových souřadnicích, rychlosti pohybu, přesnosti, atd. do řídicí jednotky robota.

3.2. Druhá metoda řízení robota

Tato metoda pracuje na principu zasílání dat z aplikace do řídicí jednotky robota, ve které běží procedury, jež daná data zpracují, vyhodnotí a podle požadavků řídí robota. Je tak větší volnost při řízení robota, jelikož lze za běhu programu upravovat jeho pohyb a okamžitě na tyto pohyby reagovat. Veškeré souřadnice jsou načítány z externích zdrojů (zde XML soubory), které jsou uloženy v počítači. Lze je tak pohodlně a operativně modifikovat dle nových poznatků a požadavků. Řídicí jednotka robota přijme data, která obsahují nejen souřadnice, na které má robot najet, ale též i řídicí kódy, které upravují jeho chování. Lze tak aktivovat kontinuální režim, kdy robot plynule najíždí přes několik definovaných mezipoloh až do cílového místa. Robot tak rychle a bezpečně dojede z jednoho měřicího místa do následujícího. Pro přesné najetí u kontroly trnem, kdy hraje přesnost velkou roli (řádky desetin milimetrů), můžeme z aplikace zasláním příslušného kódu robota zpomalit a nastavit mu s jakou přesností má dojet na dané souřadnice. Touto metodou lze přesně určit nejen, ve kterém měřicím místě se robot nachází, ale též lze monitorovat každý jeho pohyb i v rámci mezipoloh. Jistým nebezpečím je ztráta řídicích dat v rámci ethernetového přenosu či možnost nabourání se do řídicí jednotky robota a zaslání škodlivých příkazů. Toto nebezpečí však je již nad rámec zadání a je již v plné kompetenci IT administrátorů dané počítačové sítě. Další možným problémem může být nevhodnost komunikačního protokolu mezi řídicí jednotkou a aplikací a tím i možnost tak výskytu zkreslených dat, jelikož řídicí jednotka robota není primárně určena na neustálou komunikaci s aplikací a zejména na příjem a zpracování obdržených souřadnic.

3.3. Programování robota

Programování robota probíhá v dodávaném vývojovém prostředí RT toolbox (viz. Obrázek 3-4), ve kterém se píše jednoduché instrukce, ale taktéž v něm lze upravovat parametry pro robota. Je zde možnost definovat pracovní oblast robota, proudové zátěže pro jednotlivé osy (viz bezpečnost robota). Taktéž chování řídicí jednotky na různé neočekávané události.



Obrázek 3-4: Správa robota

Program pro robota je velice podobný kódu Basic, který byl a místy ještě je využíván pro jednodušší zařízení. Jelikož po programovací stránce je robotické zařízení velice jednoduché, tak i zde stačí jednodušší programovací jazyk obsahující jen základní sadu příkazů.

```
52 'subroutine for moving on position XYZ obtained by Input JEDEMP
```

```
53 *JEDEMP
```

```
54 Mov P1
```

```
55 Print #2,"100"
```

```
56 GoTo *START
```

Příkazy z aplikace do řídicí jednotky jsou posílány ve formátu *kód příkazu,P souřadnice,Jsouřadnice*. Kód příkazu můžeme též poslat samostatně bez souřadnic. Význam jednotlivých kódů je uveden v tabulce 3-1.

Tabulka 3-1: význam kódů pro robota

kód příkazu	procedura	popis
0	konec	vypne serva, zastaví program v řídicí jednotce
1	ServoOn	zapne serva
2	ServoOff	vypne serva
3	Serva	test na zapnutá serva
5	Draha	procentuální hodnota ujeté dráhy
10	ToHome	pošle robota do domácí pozice
20	SENDP	okamžitá poloha robota ve formátu XYZ
25	SENDJ	okamžitá poloha robota ve formátu J
30	JedemP	přesun pomocí instrukce MOV na souřadnice ve formátu XYZ
35	JEdemJ	přesun pomocí instrukce MOV na souřadnice ve formátu j
36	JedST	přesun pomocí instrukce MVS na souřadnice ve formátu J se stopem po aktivaci čidla (u trnu)
37	Plyn	aktivace parametru CNT – kontinuální režim
38	Nuluj	Deaktivace parametru CNT
40	Jop	převod souřadnic J na XYZ
99	Test	vrací hodnotu 99
100-199	Fin	nastavení parametru Fine (přesnost na hodnotu zmenšenou o 100
200-299	OVR	nastavení parametru OVRD (rychlost) na hodnotu zmenšenou o 100

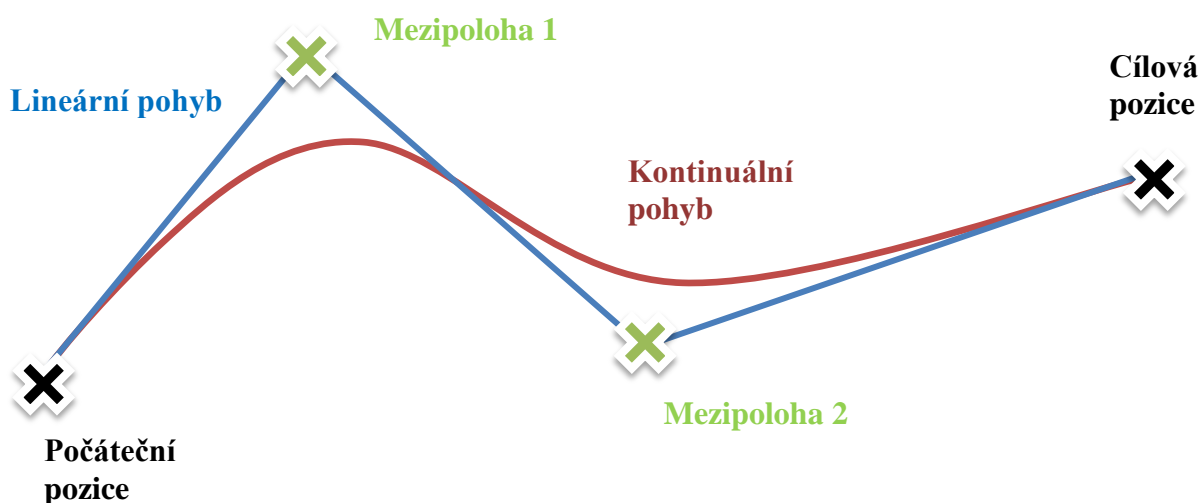
Některé příkazy budou podrobněji vysvětleny, jelikož jsou pro chod programu stěžejní a jejich přesné popsání je důležité pro korektní chod programu.

JEDEMJ, JEDEMP – řídicí jednotka zpracuje souřadnice v daném formátu a dá povel servopohonům v robotu, aby se natočily o požadovaný úhel či aby se robot nastavil do požadované polohy. Po vykonání pohybu, vrátí hodnotu 100.

JEDST – pohyb robota pomocí souřadnic v J formátu, totožné s příkazem JEDEMJ, avšak pokud je sepnuto čidlo umístěné u trnu kontrolující otvory, je robot okamžitě zastaven a řídicí jednotka navrátí aplikaci hodnotu 400. Hodnota 100 odeslána aplikací informuje, že díra není v požadovaných místech zaslepena, pokud se vrátí hodnota 400, zpracuje aplikace tuto informaci jako neprůchozí otvor.

DRAHA – vrací procentuální vyjádření již ujeté dráhy směrem k cíli. Pro polovinu dráhy je hodnota 50 %, po dojetí do požadované pozice je hodnota 100 %. Zde je ovšem velice zásadní problém, který vychází z kontinuity programu v řídicí jednotce robota. Jelikož robot vykonává příkazy v posloupnosti, ve které přijdou a další je vykonáván až po ukončení příkazu předešlého, bude tento příkaz vykonán vždy až po dojetí robota do požadované pozice, tudíž vždy vrátí hodnotu 100. Pro korektní chod tohoto příkazu by musel být příkaz spouštěn v dalším vlákně, v tzv. slotu řídicí jednotky.

PLYN – řídicí aplikace aktivuje tzv. kontinuální režim, který výrazně zkracuje dobu pro dosažení konečného místa pomocí několika mezipoloh. Pohyb je tak plynulejší bez zastavování v jednotlivých mezipolohách. Rozdíl a výhody kontinuálního pohybu lze vidět na obrázku 3-5.



Obrázek 3-5: Trasy robota

4. Kamera

Dalším důležitým prvkem celé soustavy je kamera, která vykonává vizuální a měřicí úkony. Byla zvolena Smart kamera pro výhodu zpracování dat uvnitř kamery a odesílání do počítače pouze výsledných hodnot. Tímto se celá sestava rozdělila na další nezávislou část, což umožnilo výrazně rychlejší komunikaci, zpracování dat mezi počítačem a kamerou a především urychlení samotného vývoje zařízení.

4.1. Popis

Smart kamery spojují klasickou průmyslovou kameru a výpočetní hardware (počítač) v jeden samostatný celek, který je schopen sám vyhodnocovat obraz a podle předem definovaných stavů případně ovládat jednotlivé další části systému (spouštět relé, zobrazovat status výrobku na světelných ukazatelích, atd.) Velkým přínosem je oddělení části zpracovávající obraz od řídicí jednotky pro celou výrobní linku. Do řídicí jednotky se tak dostávají pouze výstupní data od kamery obsahující velikost daného prvku v px případně i v metrických jednotkách, správně naprogramovaná kamera dokáže vyhodnotit i vizuální podobnost prvku s předlohou a poté pouze odeslat do řídicí jednotky data ve formátu Ano/Ne (True/False,...)

Smart kamera je komplexní zařízení slučující do sebe několik menších částí. Od analogové části, kterou tvoří objektiv, čočka, případně i osvětlení umístěné v kameře, přes A/D převodník, který je u snímačů CMOS možno umístit na jeden čip, čip zpracovávající obraz, po mikropočítač s RAM obsahující běžící program a Flash paměť, která je většinou vestavěna, ale existují i varianty s externí pamětí na SD kartách.

Komunikační rozhraní je určeno jednak pro konfiguraci a nahrávání programů do kamery a jednak pro komunikaci se zařízením (osvětlovače, PLC, atd...). Pro konfiguraci kamery a tvorbu programů je dobré zvolit dostatečně rychlé rozhraní, aby bylo možné pracovat s obrazem v tzv. Live režimu, kdy je aktuálně snímáný obraz ihned posílán do počítače, kde je program tvořen. V dnešní době je upřednostňováno rozhraní RJ-45 (ethernet). Toto rozhraní zvládá teoretický přenos až 1Gb/s, které je však značně ovlivňováno dalšími prvky v síti, proto je ideální postavit kameru a počítač do jedné úrovně. Dalším velice rychlým rozhraním je méně rozšířené FireWire, nabízející ve své 5pinové konfiguraci i možnost napájet kameru. Zásadní nevýhodou

tohoto rozhraní je jeho velmi malá rozšiřitelnost v počítačích a v notebookech - pouze ve své 4pinové verzi (bez napájení). Pro programování je ještě možné použít USB v revizi 2.0, toto rozhraní je ale pro kamery s vysokým rozlišením poměrně pomalé. Lze se setkat i s rozhraním Camera-Link, vytvořeném přímo pro zachytávání obrazu z kamer, či s rozhraním GigE. Jako rozhraní pro komunikaci kamery s PLC či zařízením se dále využívají klasické průmyslové sběrnice RS-232/422/485. Na tato rozhraní již není kladen tak vysoký důraz na přenosovou rychlost, ale spíše na kvalitu a spolehlivost. Posílají se většinou již jen řídicí data (rozsviť, posun,...) či výsledná data z kamery (velikost, ano/ne,...). Příklad komunikačních rozhraní na Smart kameře je zobrazen na obrázku 4-1.



Obrázek 4-1: Rozhraní kamery

4.2. Použitá kamera

Pro zařízení byla zvolena kamera od firmy Impact řady T38 (zobrazena na obrázku 4-2), která se vyznačuje vysokým rozlišením (1600x1200) s barevným CCD čipem. Volba barevné varianty byla z důvodů požadavků zadavatele na kontrolu barvy u některých dílů. Tato volba měla za následek úpravy stávajících knihoven pro ovládání kamery a značnou úpravu principu práce se získaným obrázkem. Jelikož kamera ukládá do své paměti nekomprimovaný obrázek, je jeho velikost téměř 6MB , což i po síti ethernet je veliké množství dat na poslání, tak i pro vnitřní počítač kamery. Proto při přijímání obrázku z kamery docházelo ke zpoždování, zablokování aplikace, tak i přehlcení kamery. Bylo přijato opatření, které během měření pro potřeby náhledu stahuje z kamery pouze modrou barevnou složku. Tato složka se vyznačuje tím, že je dopočítávána při zpracovávání obrazu, tj. nese menší množství informací. Takto získaný obraz se při reprezentaci jeví jako obraz v šedé tónině s rozlišením 640x480. Tím bylo potřeba přenést mezi kamerou a aplikací menší objem dat, což zaručovalo bezchybný chod zařízení. Zadavatel si však vyžádal též stahovat barevné snímky v maximálním možném rozlišení v situacích, kdy daný prvek nesplňuje požadavky na správnost (rozměr mimo tolerance, jiný tvar či barva). Pokud kamera takovýto snímek odesílala na zařízení a robot v mezičase dojel na další prvek, který byl opět špatný, a tato situace se za sebou zopakovala velice rychle, např. při měření trajektorií, kdy je na malém prostoru pořízeno několik snímků, kamera přestávala stíhat zpracovávat požadavky na odeslání snímku. V této situaci kamera začala pomalu zpracovávat požadavky, až v některých situacích přestala reagovat, stejně tak řídicí aplikace se začala přeplňovat paralelními vlákny a jelikož aplikace běžela na nevýkonném průmyslovém počítači, docházelo ke značným problémům. Byla proto zvolena výjimka při odesílání velkého obrázku pro účely archivace, robot najel v mezičase na další měřicí místo, ale zařízení neměřilo do doby, než byl daný snímek odeslán do aplikace a až poté bylo provedeno další měření. Tím se sice celé měření zpomalilo, ale jelikož chybných dílů se nebude vyskytovat mnoho a několik chybných prvků tak často za sebou, je daný ztracený čas kompenzací za případné problémy s celým zařízením.

Komunikace mezi aplikací a kamerou probíhá pomocí vzájemného posílání textových řetězců, obsahující několik slov oddělených oddělovač, zde použita dvojtečka „:“. Aplikace při odesílání do kamery nastavuje její parametry a hodnoty a kamera odpovídá výsledkem a kontrolním slovem. Kamera obdrží od aplikace údaje o IP adrese a portu, kam má kamera data odeslat. IP adresa je adresa počítače, na kterém běží řídicí aplikace. Port je odesílán z důvodu, že kamera jednak může komunikovat paralelně, ale především, že při časté komunikaci by mohl některý z portů systém Windows „obsadit či uzavřít“. Dalším odesílaným parametrem pro kameru je název programu, který se bude používat pro zpracovávání obrazu. Název programu je složen z písmene určující skupinu programu, soustřeďujících pod sebe podobné programy (pro měření průměru, vzdálenosti, vizuálních,..) a číslo přesně daného programu v dané skupině. Jednotlivé významy kódu pro skupiny jsou vysvětleny v kapitole Druhy měření. Posledním parametrem je hodnota expozice kamery při pořizování snímku, jelikož se některé programy mohou použít i pro více měření, je možné hodnotu expozice zadávat nezávisle. Po vyhodnocení snímku kamera odpovídá aplikaci ve formátu „odpověď:kód programu“. Odpovědí může být číselná hodnota daného rozměru v pixelech či text True/False u vizuálních kontrol. Kód programu je označení použitého programu, který musí být stejný jako kód programu odeslaný do kamery. Pokud by tak nebylo, je údaj vyhodnocen jako špatný.

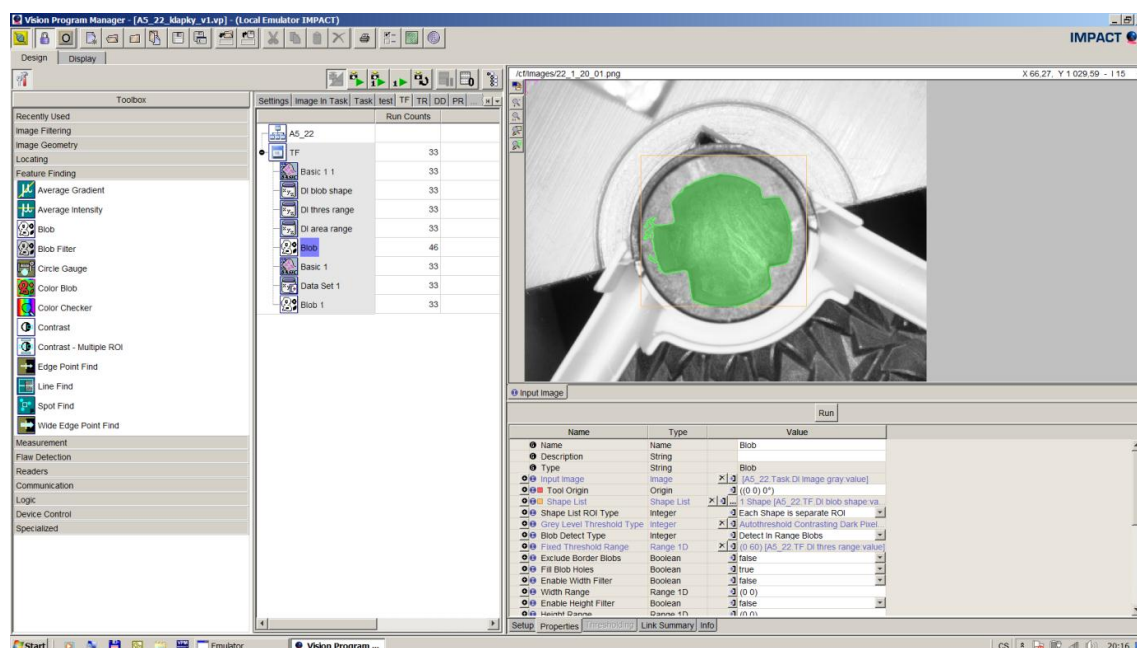


Obrázek 4-2: Smart kamera

4.3. Programování

Smart kamery se programují pomocí specifických programovacích aplikací přímo určených pro daný typ kamer (viz Obrázek 4-3). Ve většině případů se kamera učí z tzv. referenčních vzorků, pomocí nichž se nastavují, upravují, kalibrují jednotlivé kroky rozpoznávání daných parametrů (velikost, průměr, OCR, tvar,...). Při tvorbě snímků pro kameru je velice důležité zvolit správný úhel pohledu, natočení a především osvětlení. Není proto nasnadě vytvářet surové snímky bez zahrnutí okolních podmínek jako je odlesk od jednotlivých konstrukčních prvků zařízení, šum okolního osvětlení atd. Snímky se proto vytvářejí až na hotovém zařízení či v momentě, kdy již nebude zásadně ovlivněn výsledný snímek. Výhodou většiny programovacích aplikací pro kamery je simulace připojené kamery, můžeme tak vytvořit předem sadu snímků a ty poté v off-line režimu testovat na vytvořeném programu pro kameru.

Nejpoužívanějším principem vyhodnocování správného obrazu je porovnání sejmutého snímky s předem definovaným etalonem. Vyhodnocování většinou probíhá porovnáním histogramu jednotlivých obrazů buď pro celý snímek či jen pro kontrolovanou část snímku. I přes značnou jednoduchost tohoto testování je tento způsob velice účinný a spolehlivý. Ke Smart kamerám se dodává grafické programovací prostředí, v němž lze snadno definovat jednotlivé parametry pro inspekci. Mezi nejčastější operace patří detekce hran, čtení textu či měření vzdálenosti.



Obrázek 4-3: Programovací rozhraní

4.4. OCR

Optical character recognition, neboli optické rozpoznávání znaků, je metodou, která převádí tištěný (či „fyzický“) text na digitální zpracovatelný v počítači. Pro převádění tištěného textu na papíře či jemu podobného je již vytvořeno několik knihoven a algoritmů pro dekodování znaků z textu. Největším v této oblasti je ruská firma ABBYY, zabývající se tvorbou programu pro konverzi obrazu do textu od roku 1993. Pro „přečtení“ klasického textu stačí v dnešní době tento dokument naskenovat a spustit daný translační program. Ten již jednotlivé znaky převede do textového dokumentu. Požadavky na skenovaný dokument jsou velice přísné, dokument by měl být nejlépe černý na bílém, čistě vytištěn, bez rozmazaných písmen či šumu, dále by měl být skenován v rovině písma a s dostatečnou ostrostí. Těchto požadavků lze však ve většině případů snadno dosáhnout.

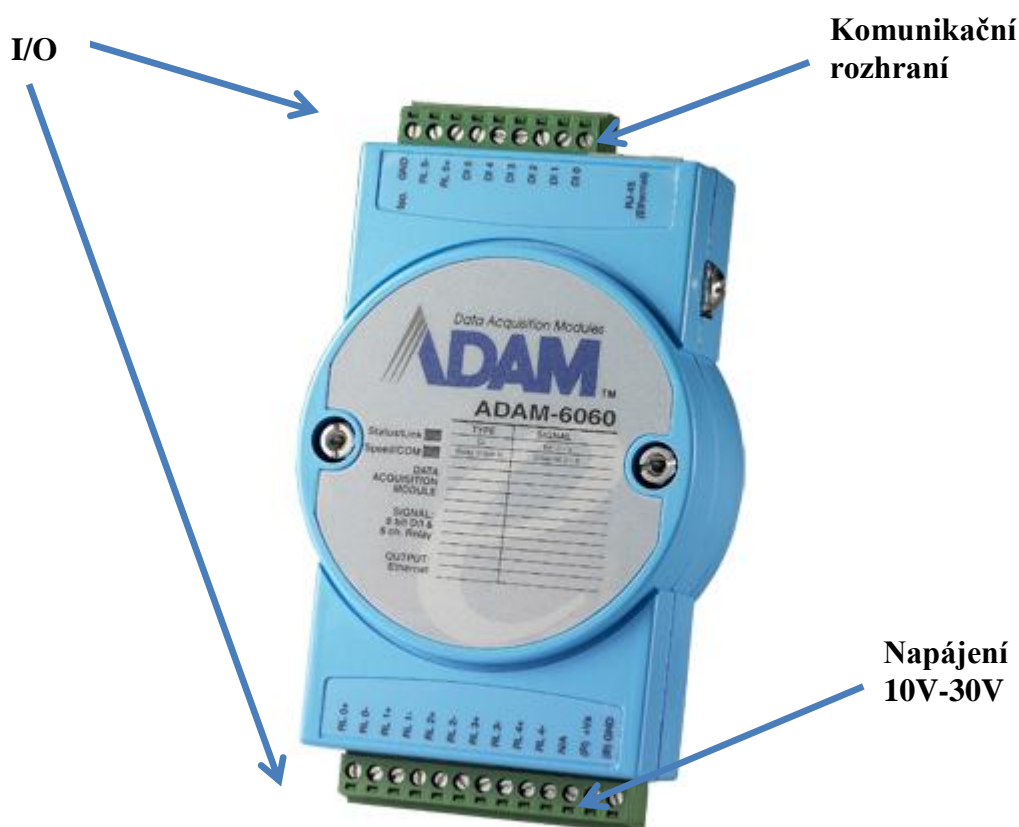
Při čtení průmyslového textu (ukázka viz Obrázek 4-4) je dodržení jakýchkoliv podmínek velice složité. Text může být různě natočen, poničen, neúplný, a dokonce i pro lidské oko nečitelný. Nejsložitějším čtením textu jsou znaky vylisované do výrobků. Text může být sice ostrý, avšak je zde velice důležité zvolit vhodné osvětlení, aby byl text rozpoznán oproti pozadí. Proto je ve většině případů nutné předem daný program „naučit“ číst dané znaky. Zde se vhodně využije, je-li k dispozici velké množství referenčních vzorků obsahující různé znaky. Po vhodném umístění textu do obrazu (nejlépe do jeho středu), správném nasvícení, nastavení potřebných parametrů (doba expozice, hloubka, prahovatelnost, ...) musíme označit jednotlivé znaky a zadat do aplikace, o jaké znaky se jedná. Aplikace si poté jednotlivé znaky uloží do své knihovny a při čtení textu vyhledává nejvíce podobný znak. Zde je velice důležité najít rozumný poměr mezi velikostí knihovny a spolehlivostí čteného textu. Obsahuje-li knihovna příliš mnoho znaků, trvá čtení textu delší dobu, avšak s jistějším výsledkem. Při menší knihovně je přečtení rychlejší, ale s nižší zárukou spolehlivosti.



Obrázek 4-4: Text na výrobku

5. Adamy

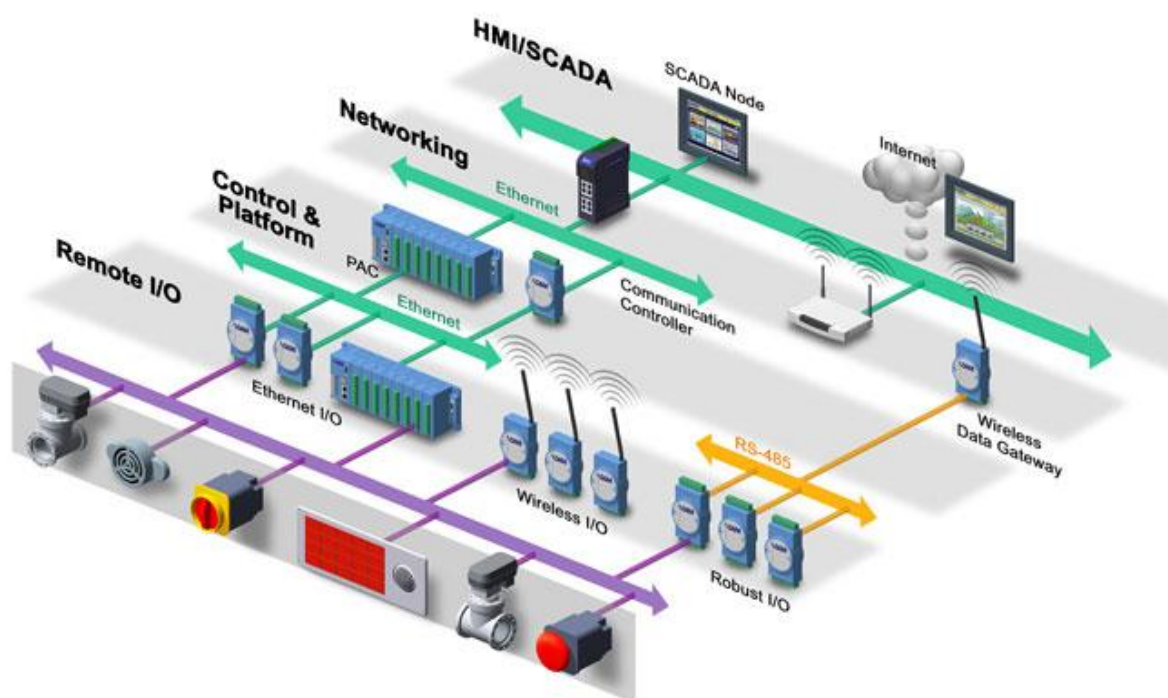
Moduly Adam od firmy Advantech jsou určeny pro sběr dat a řízení systému. Mohou být digitální či analogické, přenos dat může být po lince RS-232/485, ethernet či Wi-fi. Zařízení od společnosti Advantech se vyznačují velice kvalitním a především odolným provedením, což je předurčuje pro nasazení v průmyslovém odvětví. Jejich nastavení je díky dodávanému softwaru velice jednoduché a rychlé, a umožňuje tak pružnou reakci na změnu v řízení. Příklad modulu Adam je zobrazen na obrázku 5-1.



Obrázek 5-1: Modul Adam

Po rozšíření a standardizování ethernetového připojení nejen mezi počítači, ale i mezi průmyslovými zařízeními, je proto nejlepší možností volit zařízení Adam taktéž s podporou ethernetové komunikace. Volba ethernetové komunikace přináší velké výhody nejen v úspoře za vedení datové kabeláže, ale především možnost vzdálené kontroly Adamů. Podle firmy Advantech bude do 5let bude ethernet dominovat většině komunikačního průmyslového rozhraní. Toto umožní zmenšení a rozdělení velkých PLC strojů na několik menších smart jednotek, které budou spolehlivější, flexibilnější a v případě poruchy i lépe nahraditelné.

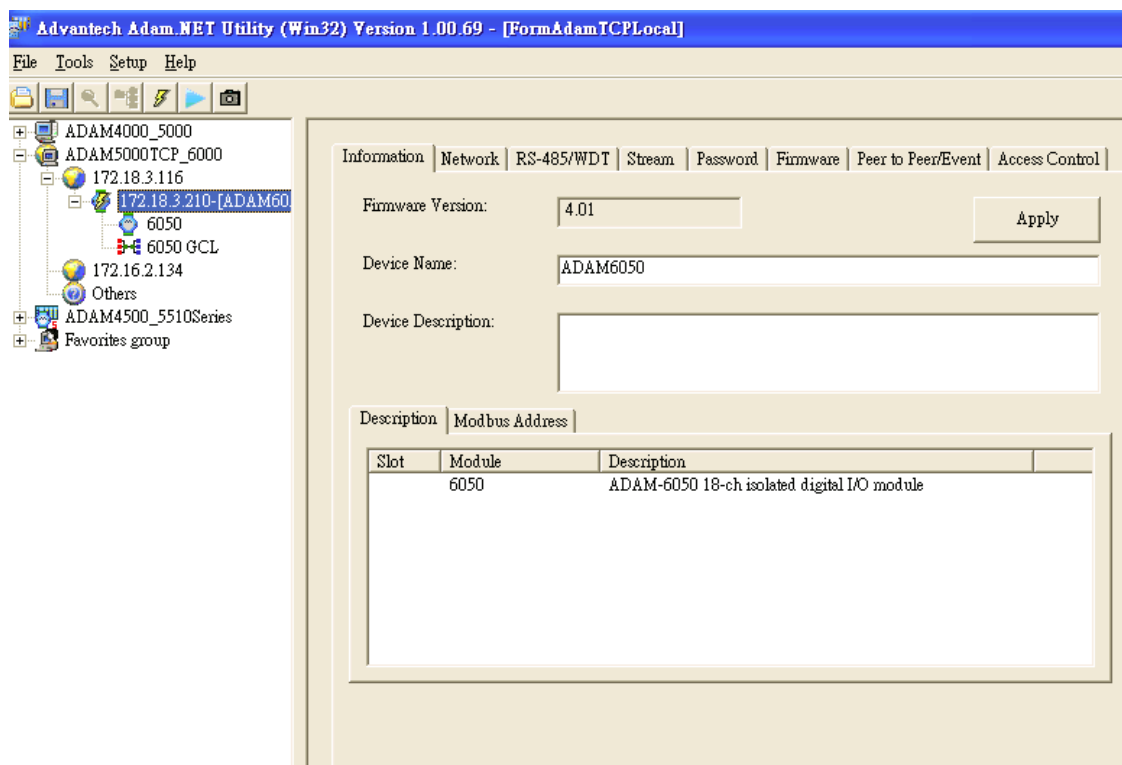
Moduly Adam řady 6000 obsahují i vlastní webové rozhraní, které lze v případě potřeby parametrizovat podle potřeb. Tímto je možnost neustálé kontroly zařízení nejen v rámci daného výrobního bloku, ale i z pozice manažerské či dohledové např. z domácí kanceláře. Lze tak operativně téměř okamžitě reagovat na jakékoliv události a situace. Pomocí jedné soustavy Adamů je možné například kontrolovat teploty v jednotlivých výměnících tepla po městě a druhou soustavou řídit výkon teplárenského bloku. Pomocí ethernetového připojení lze tak data posílat na dlouhé vzdálenosti nejen stovek metrů, ale i kilometrů. Schéma takovéto průmyslové sítě je znázorněn na obrázku 5-2.



Obrázek 5-2: Příklad využití modulů Adam

Adamy řady 6000 lze nakonfigurovat i pro „samostatnou regulaci“ soustavy. Přidáním logiky do modulu, která nastavuje hodnoty výstupu podle vstupu, může modul automaticky ovládat jednodušší zařízení. Např. pokud zaznamená čidlo příjezd vozidla, je pak vyslán signál pro otevření brány. Společnost Advantech pro programování Adamů dodává grafickou utilitu pro jednodušší tvorbu programu.

Adamy se nastavují v utilitě Adam.NET Utility, jenž je zobrazena na obrázku 5-3, vyvinuté přímo společností Advantech. Pomocí ní lze nastavovat parametry jednotlivých Adamů, monitorovat či spínat jednotlivé výstupy a nastavovat chování Adamů při určitých událostech.



Obrázek 5-3: Rozhraní pro správu modulů Adam

5.1. Řízení Adamů

V uživatelské aplikaci se Adamy řídí příkazy odesílanými po ethernetu přímo na jednotlivé moduly. Jelikož se jedná o jednotky obsahující pouze digitální vstupy a výstupy, jsou tyto porty ovládány pouze pomocí jedniček a nul. Pokud je potřeba aktivovat třetí port na daném modulu, odešle se na příslušný Adam hodnota True přímo na určitou pozici v tabulce, která určuje stav jednotlivých portů.

`adamModbus.Modbus().ForceSingleCoil(24,true);` nastaví daný port na hodnotu true

Řízení jednotlivých portů Adamu se využívá pouze u řízení pracovního osvětlení (zářivek) či při resetu robota, pokud řídíme osvětlení pro kameru, je nutné nastavit několik portů najednou. Zasíláme tak na Adamy tabulku několika hodnot určujících, které světlo se má rozsvítit, a které nikoliv. Je určena pozice prvního bytu v Adamu, od kterého se budou nastavovat porty v závislosti na přijatých datech v tabulce.

`bool[] adamy = { false, false, true, false,true, false, true, true }`

`adamModbus.Modbus().ForceMultiCoils(17, adamy);` nastaví několik portů v závislosti na přijaté tabulce

Při čtení hodnot z Adamů se čte tabulka od zadané pozice o určité velikosti. Pokud je potřeba vyčíst pouze jednu hodnotu, je určena pozice daného bytu na modulu a velikost čtené tabulky je 1.

`adamModbus.Modbus().ReadCoilStatus(iDiStart, iDiCount, out bDiData));`

`iDiStart` – pozice prvního čteného portu

`iDiCount` – počet následujících čtených portů

`bDiData` – výstupní tabulka, do které budou zapsány stavy jednotlivých portů

Adamy v tomto zařízení pracují jen v režimu pro převod počítačových dat do fyzického zařízení. Byla by zde i možnost je nahradit např. převodníkem z USB na paralelní rozhraní, které by poté spínalo výkonové tranzistory. Toto řešení by však nebylo tak robustní jako osvědčené průmyslové zařízení od společnosti Advantech.

Výhodou tohoto uspořádání je především fakt, že jej lze ovládat i z jiných míst. Pokud je potřeba aktivovat jednotlivá světla, je možnost toto provést z jakéhokoliv počítače umístěného v příslušné síti ethernet.

Veškerá komunikace s moduly Adam je prováděna přes dodávané knihovny, které je nutné implementovat do vyvíjené aplikace. V případě komunikace stačí pouze vytvořit spojení s modulem, odeslat na něj data či data přečíst a spojení řádně ukončit. Proto je práce s těmito moduly velice snadná a nic jim nebrání rozšíření i do dalších projektů.

```
AdamSocket adamModbus = new AdamSocket();

if (adamModbus.Connect(IP, ProtocolType.Tcp, 502))

{

    adamModbus.Modbus().ForceSingleCoil(24,true);

    adamModbus.Disconnect();

}
```


6. Druhy měření

Jelikož zařízení neměří jen jeden druh prvků (průměry, vzdálenosti), ale může měřit několik rozměrů, vlastností a jiných požadavků, je nutné tyto prvky vzájemně odlišit, jelikož každý prvek má jiné požadavky. Popis a kódy pro jednotlivé druhy měření jsou zapsány v tabulce 6-1.

Tabulka 6-1: Druhy měření

Typ měření	Kód měření	Program v kameře	Detailní popis
Průměr	100	d	Měření průměru
Vzdálenost	150	v	Měření vzdálenosti
Dlouhá vzdálenost	200	l	Měření dlouhé díry (na dvakrát, dva snímky)
Díra	300	--- 'r'	Měření pomocí trnu do otvoru (robot vrátí 400 při signálu od čidla, jinak vrací 100)
Vizuální	400	b	Hodnotu požadavku (True/False) zapisujeme do Pozadovano/hodnota
Trajektorie	500	t	Zkontroluje trajektorii prvku (hranu,...)
EAN kód	600	c	Čárový kód
OCR	700	o	Čtení textu (výrobní číslo,...)

Při měření rozměrů je vytvořeno 5 snímků pro průměrování s tím, že nejvyšší a nejnižší hodnota je ignorována. Tímto je zaručena velká přesnost a především stabilita měření. Pokud je vyhodnocena hodnota extrémně vysoká či nízká, je ignorována a nemá tak vliv na výslednou hodnotu. Bez této funkce se měření pohybuje pouze v jednotkách desetin milimetrů, s průměrováním bylo ovšem dosaženo značného zpřesnění a především stabilizování výsledných hodnot.

Průměr – jedná se o klasické měření průměru, kdy kamera vrací průměr otvoru či výlisku v pixelech. Otvor je detekován v určitém okolí středu, tudíž je možné používat jeden program v kameře pro více prvků.

Vzdálenost – vrací délku jistého prvku, který je definován v získaném snímku. Musí být předem určen směr (vektor) tohoto prvku, nelze tudíž používat pro každý prvek jeden program v kameře, jelikož některé prvky se měří vodorovně, jiné svisle, diagonálně,...

Dlouhá délka – tato délka je pro prvky, jejichž celá velikost se nevejde do jednoho snímku. Pokud bychom snímkovali z větší vzdálenosti, není zaručeno, že vlivem pevné ohniskové vzdálenosti, bude výsledný obraz ostrý. Aby i přesto bylo možno změřit tento rozměr, byl vymyšlen algoritmus pro kameru a robota, který dovolí tuto délku získat.

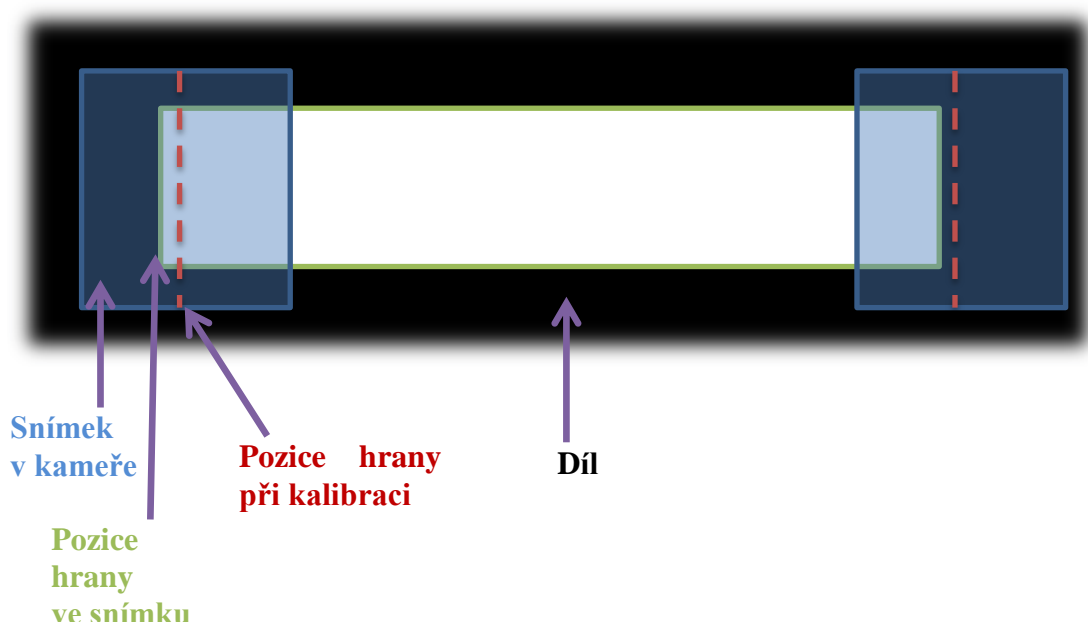
Robot najede na začátek této délky tak, aby byla hrana ohraničující danou délku uprostřed obrazu. V pořízeném snímku vyhledá tuto hranu a zjistí její vzdálenost od hrany obrazu. Tato vzdálenost je porovnávána se vzdáleností získané kalibrací změřeného předmětu. Získá se tím velikost posunutí této hrany v pixelech, kterou vynásobíme měřítkem pro převod na milimetry. Po té robot přejede na druhou hranu definující konec délky a kamera provede totéž měření. Sečtením získaného posunutí můžeme určit o kolik se daný rozměr liší od požadovaného. Přičtením vzdálenosti, kterou projede robot, se získá celková velikost dlouhé délky. Schématický princip dlouhých délek je znázorněn na obrázku 6-1.

$$1.\text{hrana} = (\text{pozice}_1 - \text{kalibrační pozice}_1) * \text{měřítko}_1$$

$$2.\text{hrana} = (\text{pozice}_2 - \text{kalibrační pozice}_2) * \text{měřítko}_2$$

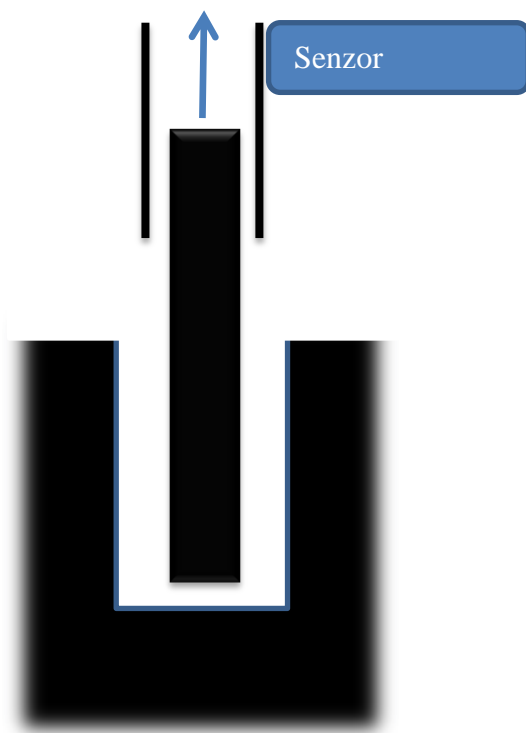
$$\text{Celková délka} = \text{kalibrovaná délka} + 1.\text{hrana} + 2.\text{hrana}$$

Pokud je daný díl správný, jen je v obrazu posunut, musí být při výpočtech jedna hrana záporná, aby se rozdíly délek vynulovaly.



Obrázek 6-1: Princip snímání dlouhých délek

Díra – díra je jediným prvkem, kdy je inspekce prováděna robotem, a nikoliv kamerou. Robot na chapadlu s kamerou obsahuje trn - Obrázek 6-2, který zjišťuje, zda je daná díra průchozí do určité hloubky. Tento trn je z pružného materiálu, aby bylo zamezeno jeho ohnutí či zlomení při neodborné manipulaci. Trn je uložen v pružném pouzdru, což umožňuje jeho posun v podélném směru. Konec trnu je snímán senzorem, který zaznamená a vyšle signál do robota v případě, že se trn v uložení posunul, což značí kontakt s předmětem. Tento signál je vyslán do řídicí jednotky robota, která okamžitě zastaví jeho pohyb, aby nedošlo ke škodám. O této skutečnosti je informována řídicí aplikace. Zda má být díra zaslepena, či nikoliv, se zaznamenává v XML v elementu *požadováno/hodnota*. V případě průchodnosti je očekávaná hodnota 100 (jelikož robot dojel až na požadovanou souřadnici v díře), pokud je potřeba neprůchodnost díry, je hodnota 400 (robot se zastavil před dojetím do bodu v díře).



Obrázek 6-2: Princip měřícího trnu

Vizuální – vizuální kontroly se provádějí porovnáváním konkrétního snímku se snímek, který byl definován jako správný. Toto porovnávání se provádí v kameře pomocí vnitřních algoritmů. Výslednou hodnotou je tak True při shodě či False pokud se snímek liší. Taktéž pomocí této funkce je kontrolována přítomnost jistých částí v daném prvku (úchyt kabeláže, díra pro nástavce,...). Očekávaná hodnota je uložena v XML jako *požadováno/hodnota*.

Trajektorie – totéž jako vizuální, jen výsledný stav se skládá z vizuálních kontrol pro více snímků. Je kontrolován obrys prvku, pořídí se několik snímku celého obrysu a tyto jsou postupně kontrolovány s očekávaným tvarem. Všechny snímky tak musí mít hodnotu True, aby byl celý prvek označen jako správný (True)

EAN – označení inspekce čárového kódu přilepeného na dílu. V kameře se dekoduje čárový kód a kamera pak odešle pouze jeho textovou prezentaci. Tento text je poté porovnán s požadovanou hodnotou. Na obrázku 6-3 je znázorněna ukázka EAN kódu.



Obrázek 6-3: Příklad EAN kódu

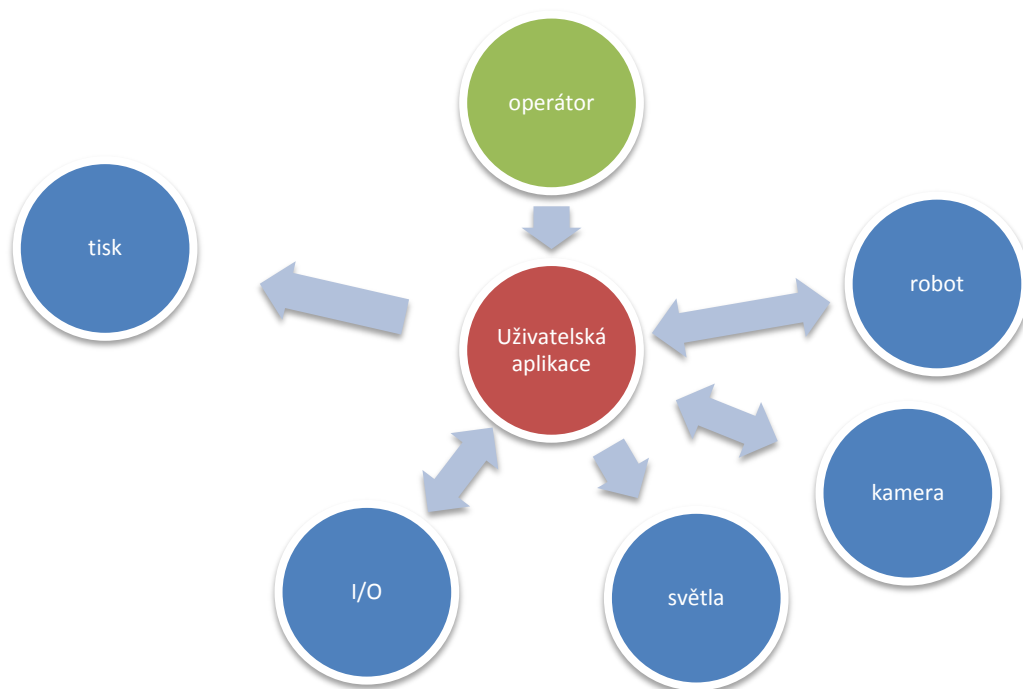
OCR – čtení textu vyraženého na výlisku. Tento text je kamerou přečten a porovnán s očekávanou hodnotou. Pokud některý ze znaků není přečten, je nahrazen „?“.

Podrobnější popis viz OCR.

7. Uživatelská aplikace

Uživatelská aplikace je spojením mezi operátorem (uživatel) a sestavou. Komunikuje s jednotlivými zařízeními a odesílá jim žádosti o jednotlivé operace, zobrazuje uživateli nabídku dílů, vkládání informací, změření jednotlivých dílů a tisk výstupního protokolu.

Vazby na uživatelskou aplikaci jsou znázorněny na následujícím obrázku 7-1.



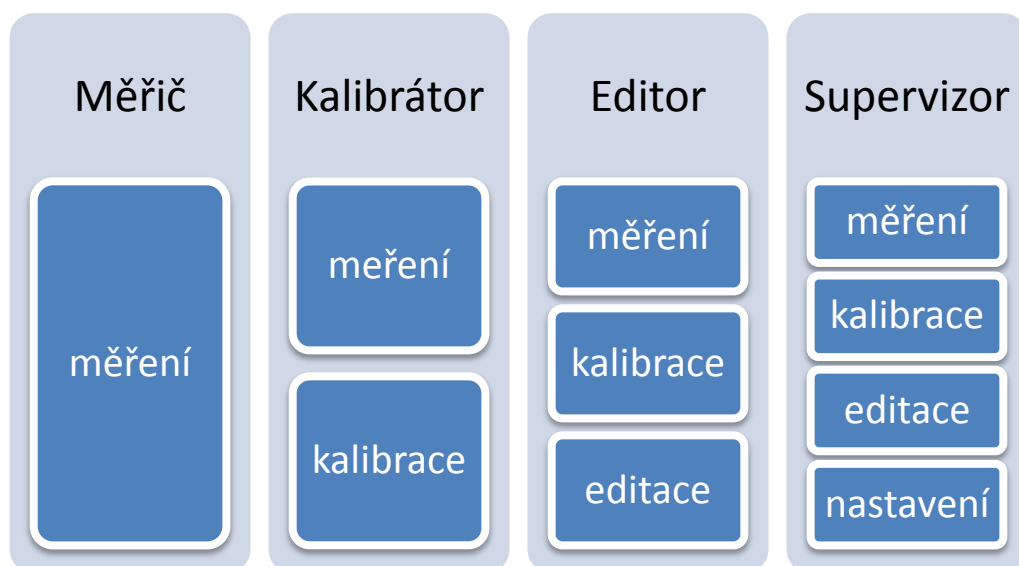
Obrázek 7-1: Struktura uživatelské aplikace

Hlavní vstupní jednotkou je operátor, který zadává požadované operace do uživatelské aplikace, která je poté vykonává řízením ostatních jednotek. Operátor, tak nepřistupuje přímo k robotovi či ke kameře, ale pouze ke grafickému rozhraní uživatelské aplikace. Tím je řízení velice intuitivní a operátor tak nemusí složitě nastavovat jednotlivé I/O, zapínat potřebné osvětlení, či zadávat příkazy do kamery. Veškeré tyto operace vykonává uživatelská aplikace, která operátorovi zobrazuje jejich aktuální průběh.

Aplikace běží na průmyslovém počítači se 17“ dotykovým displejem, proto je aplikace uzpůsobena pro základní práci bez použití myši. Z toho důvodu jsou používána rozměrná tlačítka pro snadnější obsluhu.

7.1. Stupně přidělených práv

Každý operátor má přiřazen tzv. stupeň oprávnění (viz. Obrázek 7-2), který zaručuje, že složitější či systémové operace mohou vykonávat pouze proškolení a zkušení operátoři. Zamezí se tak přístupu laiků do nastavení IP adres pro jednotlivé jednotky, souborových adresářů s XML soubory či s výsledky.



Obrázek 7-2: Schéma oprávnění

7.1.1. Přihlašovací obrazovka

Obrázek 7-3: Úvodní obrazovka

Na obrázku 7-3 je znázorněné přihlášení uživatele s oprávněním pro editaci údajů, tento stupeň mu umožňuje měřit díly, kalibrovat je a upravovat informace o nich, jako jsou názvy dílů, popis nápovědy, čísla dílů, SAP číslo, tolerance hmotnosti, povolení a zakázání měřit jednotlivé prvky, či zadávat toleranci pro požadované naměřené hodnoty (viz Obrázek 7-4). Uživatel však nemůže upravovat systémové údaje skrývané se pod tlačítkem Admin.

Editujte informace

Název: Testovací díl
Číslo: 1254997A
SAP: e5417B
Projekt: 98b přídavná osvětlovací část
Materiál: rs 85% TTuX
Hmotnost: 1438.2 g ± 43,1 g

Měřené prvky
☐ pro všechny díly levá číslo 998831D
☒ vybrat vše

120	100.02	100.07
230	100.05	100.08
170.01	100.06	
170.02	200	
100.03	100.04	

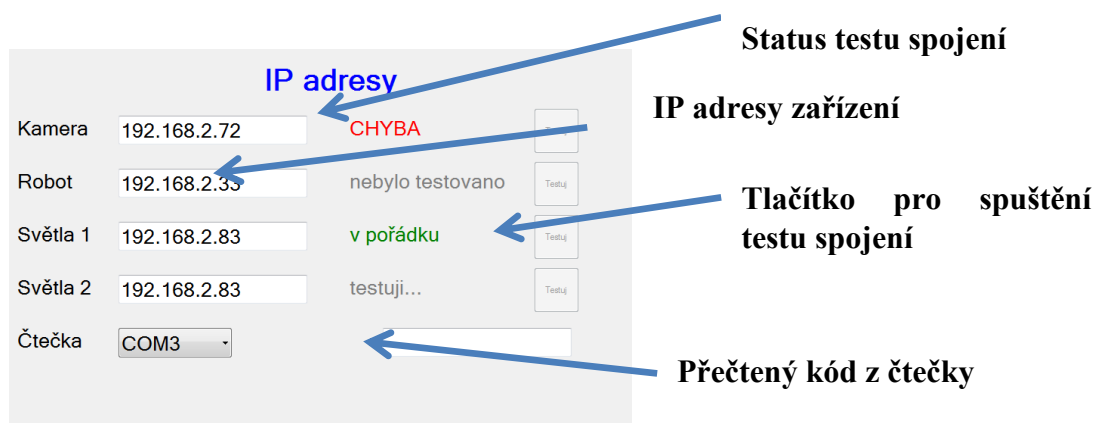
Vložte tolerance

část	znak	dt	střední	ht	rozsah
levá	120	0	8.5	0.1	8,500 - 8,600
levá	230	0	6.5	0.2	6,500 - 6,700
levá	170.01	-0.4	183.2	0.4	182,800 - 183,6...
levá	170.02	-0.4	183.2	0.4	182,800 - 183,6...
levá	200	-0.85	194.83	0.85	193,980 - 195,6...
pravá	220	-0.7	192.09	0.7	191,390 - 192,7...
pravá	210	-0.7	182.94	0.7	182,240 - 183,6...
pravá	130	0	8.5	0.1	8,500 - 8,600
pravá	60.01	0	8.5	0.1	8,500 - 8,600
pravá	60.02	0	8.5	0.1	8,500 - 8,600

Obrázek 7-4: Editace informací a tolerancí

7.1.2. Supervizor

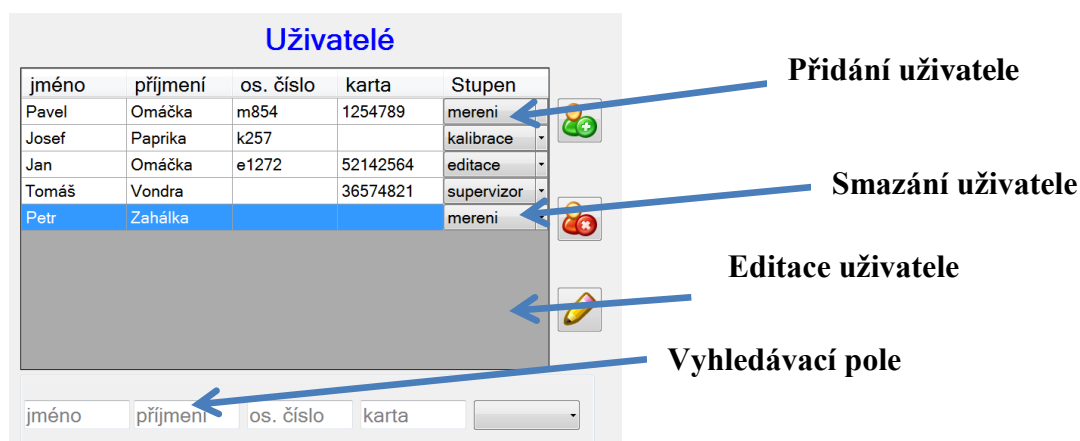
Uživatel s oprávněním supervizor může provádět veškeré operace, především měnit nastavení systémových údajů. Lze tak nastavit, do jakého adresáře se budou ukládat soubory obsahující výsledky z měření a snímky špatných prvků, adresář pro načítání vstupních XML souborů a další. Supervizor může při změně počítačové sítě upravovat IP adresy jednotlivých zařízení a poté provádět jejich test spojení s počítačem (viz Obrázek 7-5). Lze taktéž nastavit port pro čtečku přístupových čipů a provést její test. Po správném nakonfigurování čtečky (nastavení správného sériového portu) a jejím přiložením se zobrazí přečtený kód. Testování kamery probíhá zasláním testovacího řetězce, na který kamera odpoví textem „True“, pokud se tak nestane, je tento pokus 5x zopakován a poté zapsán status „chyba“. U robota testování probíhá obdobně, pouze na testovací příkaz odpovídá hodnotou 99. Test světel probíhá vytvořením spojení mezi počítačem a příslušným Adamem, pokud je spojení v pořádku, je i test v pořádku. Při testech Adamů se neodesílají žádné řetězce, jen se testuje vznik spojení.



Obrázek 7-5: Testování spojení

Supervizor má právo přidávat a odebírat uživatele a též jejich editaci. Rozhraní pro jejich správu je zobrazeno na obrázku 7-6. Každý uživatel má své unikátní osobní číslo a kód přihlašovacího čipu. Osobní číslo slouží pro interní správu uživatelů a mělo by být unikátní pro každého zaměstnance. Kód čipu musí být uložen pouze u osob mající přístup k zařízení, tím se povoluje jejich přihlášení a provádění operací.

V tabulce obsahující seznam uživatelů lze filtrovat zobrazené uživatele pomocí vyhledávacích polí umístěných ve spodní části. Po jejich vyplnění se automaticky zobrazí pouze uživatelé vyhovující daným požadavkům. Zadaním více parametrů lze filtrování více upřesnit. Mohou být tak například zobrazeny veškeré osoby se stupněm *editace* s křestním jménem obsahujícím písmeno *J*.



Obrázek 7-6: Správa uživatelů

Pavel Omáčka – uživatel s pověřením práce na zařízení s pravomocí měřit

Josef Paprika – uživatel bez přístupu k zařízení (nemůže se přihlásit pomocí čipové karty)

Jan Omáčka – uživatel s přístupem a s hodnotí editace

Tomáš Vondra – uživatel s přístupem jako supervizor, ale bez os. čísla (již nepracuje ve firmě, mělo by se smazat i číslo karty)

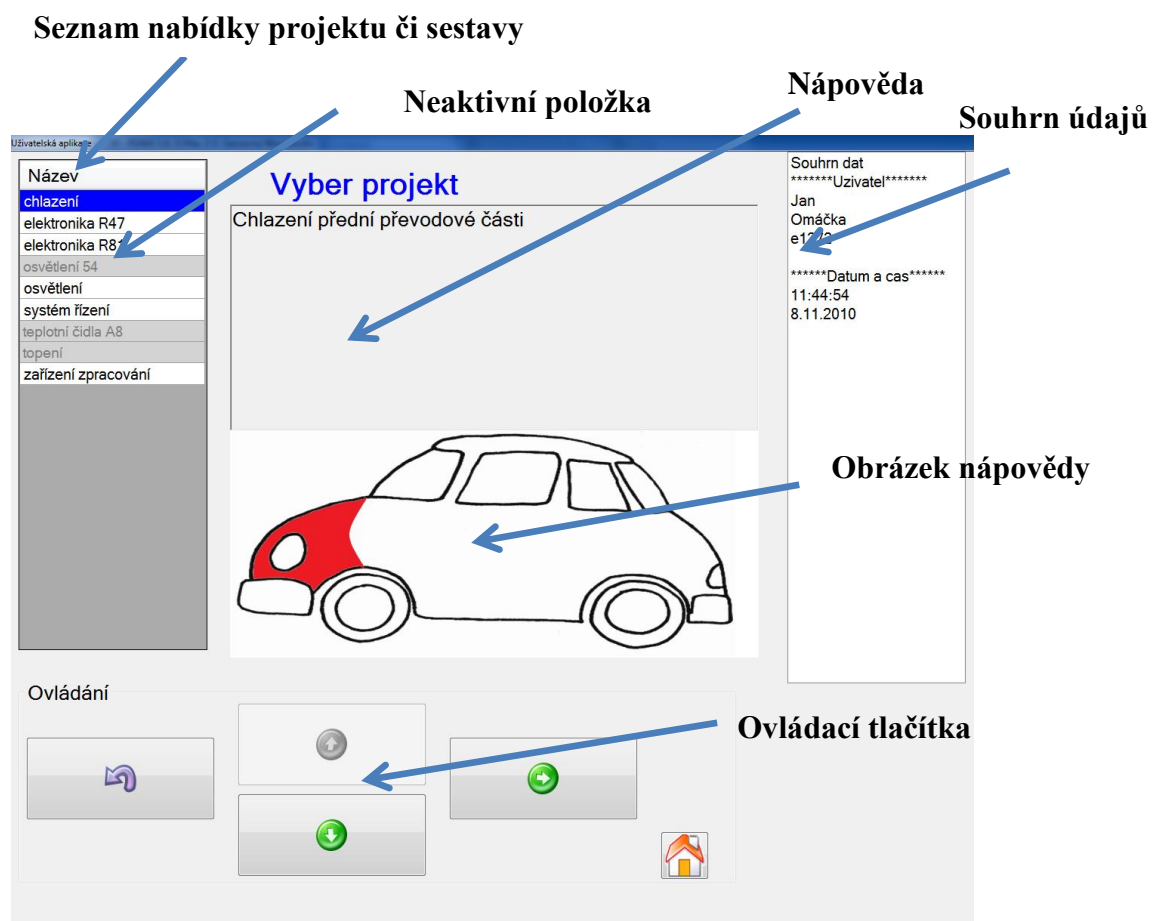
Petr Zahálka – korektně smazaný operátor, který již ve firmě nepracuje, tudíž je smazána i jeho čipová karta

Pokud při přidávání či editaci uživatele nastane situace, že dané osobní číslo či kód karty již používá jiná osoba, je o tom operátor vyrozuměn a je mu nabídnuta možnost tuto osobu zobrazit či vymazat kolidující údaje. V momentě, kdy jsou veškerí uživatelé smazáni, či je poničen soubor nesoucí informace o uživatelích, lze se k zařízení přihlásit s jakýmkoliv čipem jako supervizor. Byla by možnost i povolení přihlášení pomocí tzv. červeného čipu, který je v programu napevno zanesen a který by byl k dispozici pouze hlavnímu správci zařízení.

7.2. Výběr projektu a dílu

Operace měření, kalibrace či editace údajů je prováděna vždy pro konkrétní díl, proto je nutné před samotným úkonem daný díl správně vybrat pomocí nabídky. Pro snadnější orientaci je též nabízen stručný popis a náhled každého projektu či dílu. Projekt soustřeďuje několik dílů pod jednu hlavičku.

V průběhu zavádění systému do výroby bylo zjištěno, že existuje několik výrobců jednotlivých forem, které se v některých detailech zásadně liší. Hlavní odlišnost byla v místě vylisování čísla výrobku, tudíž robot s kamerou musí vědět, na kterém místě je dané číslo, tj. kdo formu vyrobil. Je proto nutné po výběru dílu vybrat i výrobce dané formy, tímto výběrem se určí, který XML soubor se má načíst se správnými souřadnicemi pro korektní pozici čísla. Popis obrazovky pro výběr projektu je popsán na obrázku 7-7.



Obrázek 7-7: Výběr projektu

Výběr pro konkrétní díl probíhá totožným způsobem.

7.3. Vložení informací

Po výběru požadovaného dílu je nutné zadat potřebné vstupní informace o daném dílu zobrazené na obrázku 7-8. Operátor musí zadat, o jaký vzorek v sérii se jedná, zda první, poslední či přeseřizení. Údaje jako jméno, datum, název a číslo dílu se získávají z XML dat či z databáze. Dva údaje informující o čísle dílu znamenají, že díl se skládá z více částí (horní/dolní či levá/pravá) a pro každou část je nutné mít vlastní číslo. Tato situace nastane i v momentě, kdy se najednou kontroluje více stejných menších dílů.

Číslo výlisku se kontroluje během identifikace, aby byla jistota, že bude změřen správně zadaný díl. Údaj je načítán přímo z hlavičky XML souboru, program je však připraven i pro ruční zadávání čísla přímo v okně „vložte informace“ a toto vložené číslo bude poté porovnáváno s číslem přečteným z výrobku.

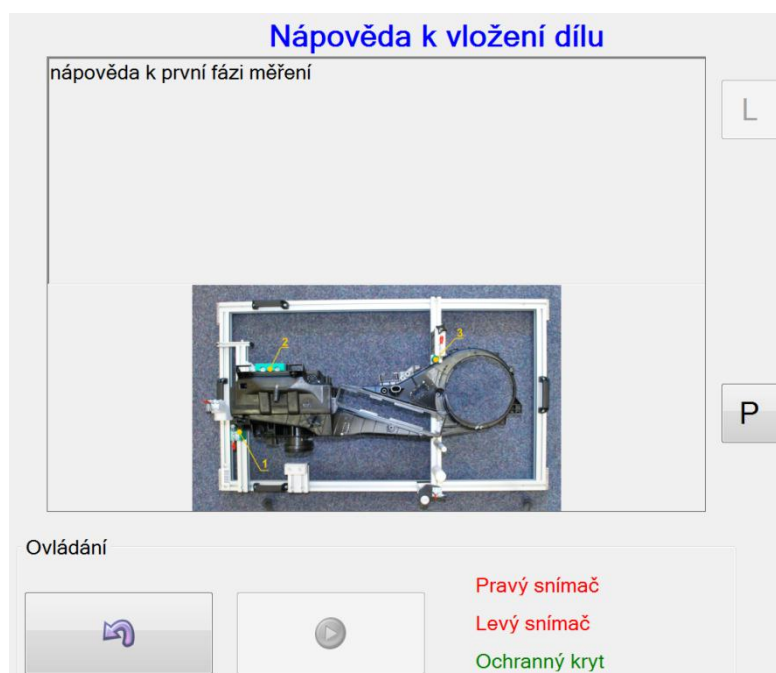
Dalším údajem zadávaným operátorem je hmotnost daného dílu, která by měla být v předem určené toleranci pro každý díl. Pokud je opět najednou měřeno více menších dílů, je nabídnuto vložení hmotnosti pro každý prvek zvlášť. V rozbalovací nabídce jsou seznamy lisů pro daný díl, tyto lisy jsou spárované pro každý díl, který mohou vyrobit, tímto je nabídka menší, protože neobsahuje lisy, na kterých by daný díl nebylo možné vyrobit. Poslední volitelnou položkou je možnost vložení poznámky k danému dílu, popisující měřený prvek či jiný text, který je později zobrazen na výstupním protokolu.

Vložte informace	
	První
jméno	Jan Omáčka
Datum	8.11.2010 - 13:46:07
název	testovací díl
číslo	998831D 998830Q
hmotnost	1395,100-1481,300g
číslo lisu	159.12
Poznámky	vložte poznámku k projektu

Obrázek 7-8: Vkládání informací

7.4. Náповѣда

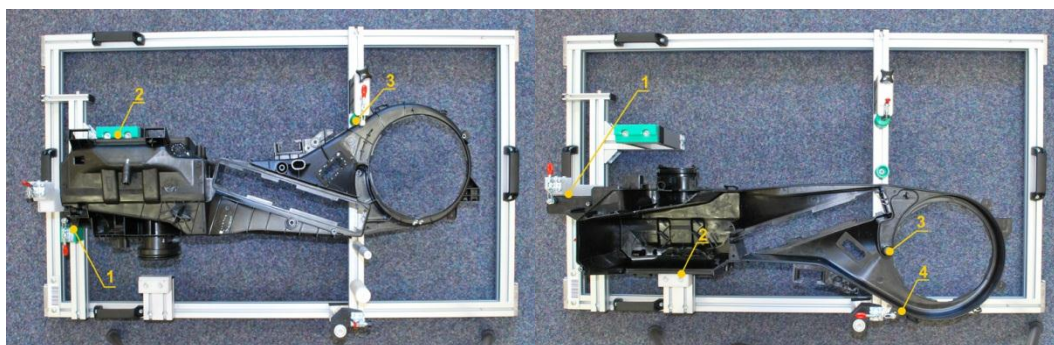
Po vložení veškerých potřebných údajů a informací je aplikace připravena měřit daný díl. Nejprve je operátorovi zobrazena nápověda a obrázek pro správné vložení přípravků a dílů (viz. Obrázek 7-9), kde je názorně zobrazeno, jak daný díl uložit a na jaký přípravek. Korektní uložení dílu je velice důležité pro výsledné měření, jelikož posunutí dílu může způsobit, že robot nenajede nad správný prvek, který je tím špatně vyhodnocen. Horší situace nastane při měření díry, kdy robot musí najet s přesností na desetinu milimetru, jinak by mohlo hrozit poškození trnu. Jelikož do zařízení je možné vložit dva rošty najednou, je umožněno přepínání mezi nápovědy pro levou a pravou stranou pomocí tlačítek. Nápověda je vztažena k aktuálně měřené fázi, protože je nutné mezi nimi otáčet díly, aby bylo možné měřit i prvky z druhé strany, kam se robot s kamerou nedostane.



Obrázek 7-9: Náповѣда

Před aktivací tlačítka „start“ je nutné zajistit veškeré bezpečnostní prvky, správné ukotvení roštů v zařízení, které je snímáno senzory, a zavření ochranného krytu bránící styku operátora s pracovním prostorem robota. Tyto signály jsou přenášeny pomocí digitálních vstupů v modulech Adam.

Rošty jsou upínací podložky, na které se vkládají jednotlivé měřené díly. Pro každou sadu dílů je nutné vytvořit vlastní rošty, jelikož se díly upínají v rozličných polohách a za různé části. Při tvorbě roštů je tak nutné zohlednit všechny prvky v dané sestavě, aby bylo možné použít jeden rošt pro více podobných dílů. Dále je nutná velice úzká spolupráce při jejich návrhu s potřebami inspekce jednotlivých prvků, aby bylo možné s robotem a kamerou dosáhnout potřebných pozic a nedocházelo ke kontaktu s upínacími či konstrukčními prvky. Podoba roštů i s upnutými díly a očíslovanými úchytnými místy je na obrázku 7-10.



Obrázek 7-10: Podoba roštů

7.5. Inspekce dílu

Inspekce dílu se skládá ze tří postupných částí, které zaručí korektní měření správného dílu. Jelikož operátor se může zmýlit při vkládání dílu, ať již vložením jiného dílu, než který je vybrán v databázi, či jeho špatné založení. Každý z těchto omylů by mohl být fatální pro kameru, měřicí trn, výlisek či celé zařízení. Všechny tři části jsou rozepsány v následujících kapitolách.

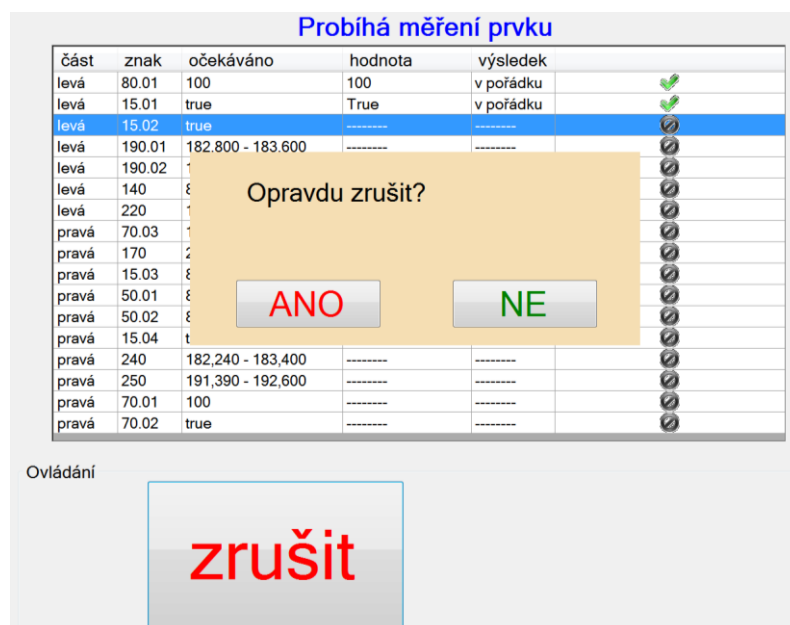
O bezpečí operátora se starají ochranné dveře, které oddělují prostor mezi robotem a okolím. Robot se nespustí bez jejich uzavření. Pokud však operátor z nějakého důvodu tyto dveře otevře, robot se okamžitě zastaví a je o této skutečnosti informován kontrolkou na řídicím panelu. Běh aplikace se zastaví a zobrazí se varovná hláška, kterou je nutné potvrdit přihlášení operátora s nejvyšším oprávněním. O veškerých neočekávaných událostech jako je odpojení kamery, robota či modulů Adam, nekorektní data od kamery (místo čísla vrátí True/False), a další, informován operátor chybovou hláškou, kterou může odblokovat pouze supervizor. Všechny tyto chyby jsou zaznamenány do logu pro pozdější identifikaci.

Před každým začátkem inspekce vyjíždí robot z domácí pozice po předem určené trase k prvnímu měřenému místu. Tuto trasu zadáváme pomocí mezipoloh, aby robot bezpečně dojel na danou pozici bez ohrožení sebe či výrobku. Přes mezipolohy projíždí robot v tzv. kontinuálním modu pro rychlý a plynulý pohyb, na koncové místo dojíždí s patřičnou přesností, dle měřeného prvku. Při měření vizuálních parametrů (tvar, obrys) či rozměrů vzdálenosti a průměru je možné snížit přesnost robota, jelikož kamera dokáže pracovat s určitou tolerancí posunutí objektu v obrazu. Po tomto najetí je nutné daný prvek správně nasvítit pro získání nejideálnějšího obrazu. Správné nasvícení dokáže velice ovlivnit výsledné měření v kameře. Při špatném osvětlení ani nejlepší program nezaručí 100% úspěšnost. Světla jsou ovládána pomocí modulů Adam, na které aplikace pošle žádost o sepnutí relé na daných výstupech. Po rozsvěcení světel pořídí kamera snímek, na kterém poté vyhodnocuje potřebné údaje. Po pořízení snímku se světla zhasnou. Pokud by však z nějakého důvodu kamera neodpověděla, že je snímek pořízen, zůstala by světla rozsvěcena a hrozilo by jejich poškození. Proto byla do aplikace zabudována ochrana světel s jejich automatickým zhasnutím po 10 sekundách od rozsvěcení. Při měření rozměrů je vytvořeno 5 snímků pro průměrování

s tím, že nejvyšší a nejnižší hodnota je ignorována. Tímto je zaručena větší přesnost měření.

U vizuálních kontrol (trajektorie, obrys) se provádí měření pouze jedno. Při měření tzv. dlouhých délek, kdy se obraz nevejde celý do objektivu kamery, jsou rozpoznávány hrany jednotlivých konců rozměrů, které jsou porovnány s ideálním umístěním hrany při její kalibraci. Bližší popis jednotlivých principů měření je v kapitole Druhy měření

Pokud chce operátor korektně ukončit inspekci dílu, při kterém není nutné použít tlačítko nouzového zastavení, je možné přerušit měření tlačítkem v ovládací aplikaci. Po stisknutí tohoto tlačítka s nápisem „zrušení“ je operátor požádán, aby tento požadavek potvrdil (viz. Obrázek 7-11), z důvodu zamezení nechtěného zmáčknutí. Po potvrzení své volby dokončí zařízení měření aktuálního prvku a vrátí se přes bezpečné body do domácí pozice. Pokud je robot aktuálně v pohybu, dokončí dojezd až do měřicí polohy, ve které jsou již zadány bezpečné body, přes které se může robot vrátit do domácí polohy. Z mezipoloh se robot nemůže vracet domů, jelikož pro ně není nastavena trasa, po které se může bezpečně vracet.

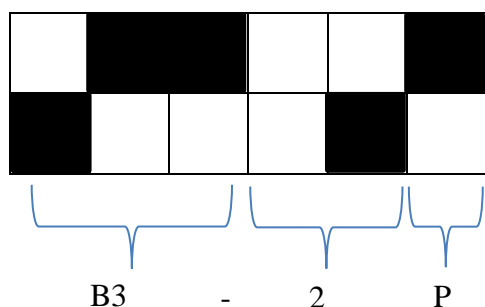


Obrázek 7-11: Zrušení měření

7.6. Kontrola

Před započítím samotného měření, především před přiblížením robota k jednotlivým dílům, je nutné ověřit, zda je vložen správný díl, resp. některý díl z dané sady. Toto zjištění se provádí pomocí přečtení unikátního kódu na jednotlivých rostech. Tento kód zaručí, že byl vložen korektní rošt s dílem. Pro každou sadu dílů, které si jsou velice podobné a liší se jen nepatrně, je určen jeden rošt pro upnutí. Pokud je vložen daný rošt a souhlasí číslo roštu s očekávaným číslem, je zaručeno, že vložený díl je podobný vybranému dílu a robot se tak k němu může přiblížit.

Každý rošt dostal unikátní systém kódování a jeho prezentaci do obrazového kódu pro čtení kamerou. Prvotně bylo označení roštů převáděno do čárového kódu (EAN) a ten poté při kontrole čten. Toto řešení však vykazovalo nespolehlivost, jelikož jemný čárový kód byl velice náchylný na zašpinění a ani poloha kamery nebyla ideální pro jeho jisté přečtení. Byl proto vytvořen unikátní kódovací systém, který dokázal jednoduché označení převést do binární matice, a který je znázorněn na obrázku 7-12.



Obrázek 7-12: kódování roštů

B3 – tato hodnota označuje vybraný projekt. Kód je napevno zadán, tudíž změnou hodnot (pozic černých čtverečků) vrátí kamera, že kód nebyl rozpoznán

2 – hodnota označující rošt v daném projektu pro určitý díl či díly. Tato hodnota je automaticky vypočítána. Kód je brán jako binární číslo a to poté převedeno na decimální hodnotu. Zde 0010 -> číslo 2. Je k dispozici 2^4 hodnot, tj. 16

P – určuje stranu, na kterou se rošt ukládá (pravá, levá). Každý čtvereček určuje jednu stranu, smí tudíž být vybrán právě jen jeden.

Další kontrolou je ověření vložení dílu ve správné „fázi“, jednotlivé fáze označují otočení dílu vrchem či dnem vzhůru. Každá strana se ukládá do specifických pozic na roštu, které jsou v jiných místech roštu. Je tak možné zkontrolovat i správné natočení dílu. Kamera zjistí, na které straně roštu je díl uložen, tím je vyhodnoceno, ve které fázi je díl a tím předejít možnému poškození zařízení či dílu. Průběh kontroly se zobrazuje na monitoru počítače a její podoba je na obrázku 7-13.

Probíhá kontrola správného vložení

část	znak	očekáváno	hodnota	výsledek	obrazek
levá	K1	true	True	v pořádku	
levá	K2	123	123	v pořádku	
pravá	K1	123	123	v pořádku	
pravá	K2	true	-----	-----	

Ovládání

zrušit

Souhrn dat
Připojování....
Připojeno
najezd na bod levá:K1
True:b2
ID:K1->v pořádku
najezd na bod levá:K2
123:c2
ID:K2->v pořádku
najezd na bod pravá:K1
123:c3
ID:K1->v pořádku
najezd na bod pravá:K2

Obrázek 7-13: Kontrola vložení

7.7. Identifikace

Po kontrole správnosti uložení dílu se provádí identifikace, zda je vložen korektní díl, jelikož některé díly se liší jen nepatrně, jiným tvarem některých prvků, jejich nepřítomností, či číslem vyraženém na dílu. Touto identifikací je naprostá jistota vložení dílu, který byl vybrán z nabídky, aby se předešlo nekorektnímu měření či poškození.

Při čtení čísla z dílu nastává příležitostná chyba, jelikož číslo nemusí být správně vyraženo, může se objevit splynutí některých cifer, či jejich poničení v rámci manipulace s dílem. Z tohoto důvodu je nutné ošetřit tuto událost při špatném přečtení čísla, kamera při nepřečtení některé z cifer vrátí místo této cifry znak otazníku '?'. Program takto detekuje nekorektnost čtení a nabídne operátorovi, aby daná čísla doplnil (viz. Obrázek 7-14). Ten musí otevřít zařízení a ručně přečíst číslo z dílu, které poté dopíše do aplikace. Pokud je toto číslo správné s očekávaným, je možno přistoupit k samotnému měření.

Probíhá kontrola prvku

část	znak	očekáváno	hodnota	výsledek	obrazek
levá	I3	true	True	v pořádku	✓
levá	I8	123	12?	doplnit	⚠
levá	I4	true	True	v pořádku	✓
pravá	I1	true	True	v pořádku	✓
pravá	I2	123	123	v pořádku	✓

Ovládá

Číslo výlisku bylo špatně přečteno
prosím opište jej

díl levá, prvek I8

Obrázek 7-14: Identifikace dílu

7.8. Měření

Při probíhající měření vidí operátor okamžitou hodnotu získanou z kamery po přepočtu na metrické jednotky (mm) a ty jsou ihned porovnány s očekávanými hodnotami s tolerancí udanou v XML souboru. Pokud se změřená hodnota vychyluje od daného rozmezí, je tento prvek označen jako špatný, jelikož neodpovídá toleranci. U vizuálních měření očekáváme hodnotu True či False a i ta musí přijít od kamery, jinak je taktéž prvek označen jako nevyhovující. U trajektorie, kdy je vizuálních měření více pro jeden prvek (kontrola obrysu se může skládat např. i z pěti snímků), musí výsledný snímek odpovídat očekávané hodnotě, u trajektorií je tato hodnota vždy True. Výsledná hodnota při měření díry může odpovídat číselnému údaji 100 či 400. Hodnota 100 znamená, že robot projel celou trasu v daném otvoru a nenajel hrotem na žádnou překážku, tudíž je díra do dané hloubky průchozí. Pokud však narazí na překážku a měřící hrot sepne, je vyslána hodnota od robota 400 (ukončení trasy) a uživatelská aplikace obdrží formaci, že díra je neprůchozí. Tyto hodnoty jsou opět porovnány s očekávanými, jelikož u některých děr je požadována průchodnost, u některých nikoliv. Podoba průběhu měření je znázorněna v obrázku 7-15.

Probíhá měření prvku					
část	znak	očekáváno	hodnota	výsledek	
levá	80.01	100	100	v pořádku	✓
levá	15.01	true	True	v pořádku	✓
levá	15.02	true	True	v pořádku	✓
levá	190.01	182,800 - 183,600	184,3	špatné	✗
levá	190.02	182,800 - 183,600	183,1	v pořádku	✓
levá	140	8,500 - 8,600	8,5	v pořádku	✓
levá	220	193,980 - 195,680	194,86	v pořádku	✓
pravá	70.03	100	100	v pořádku	✓
pravá	170	2,100 - 2,300	2,2	v pořádku	✓
pravá	15.03	8,500 - 8,700	8,6	v pořádku	✓
pravá	50.01	8,500 - 8,600	8,6	v pořádku	✓
pravá	50.02	8,500 - 8,600	11	špatné	✗
pravá	15.04	true	-----	-----	⊗
pravá	240	182,240 - 183,400	-----	-----	⊗
pravá	250	191,390 - 192,600	-----	-----	⊗
pravá	70.01	100	-----	-----	⊗
pravá	70.02	true	-----	-----	⊗

Obrázek 7-15: Měření dílu

7.9. Výstupní protokol

Každé ucelené měření se ukládá do samostatných XML souborů, které lze po skončení nechat vytisknout. Ve výstupním protokolu lze nalézt informace o operátorovi provádějícím měření, typ měření, datum a čas, informace o dílu a především výstupní hodnoty pro každý změřený prvek.

Pro pozdější kontrolu správnosti dílu jsou první v seznamu prvků zobrazeny identifikační prvky a jejich očekávané a změřené hodnoty. V každém měřicím protokolu musí mít tyto identifikační prvky správné hodnoty, jinak není zaručeno, že byl změřen správný díl. Program nepřipustí k měření nesprávné díly, ale o této kontrole by nebyl veden záznam, že byla skutečně provedena, proto jsou i identifikační prvky zaznamenány do výstupního protokolu.

Ke každému prvku lze zapsat v XML poznámku, která je poté vytištěna na protokolu a může přesněji charakterizovat daný prvek. U rozměrových prvků je uvedena i požadovaná hodnota s tolerancí, pokud je výsledná hodnota mimo tuto toleranci, je výsledek graficky odlišen. Je-li zakázáno daný prvek měřit, je ve výstupním protokolu nahrazena výsledná hodnota znaky “-----“.

Výstupní protokol z měření je v příloze B – Výstupní protokol.

8. XML

XML, anglicky eXtensible Markup Language - rozšiřitelný značkovací jazyk, je v počítačovém oboru velice hojně využíván, především v prostředí internetových stránek jako hlavní zdroj informací a uložení dat. V tomto formátu mohou být data přenášena mezi počítači, aplikacemi, servery. Data psaná v XML kódu obsahují pouze čistá data bez grafického rozhraní, tudíž zobrazení těchto dat v různých aplikacích bude mít za následek i jejich různou prezentaci, avšak obsah bude stále stejný. Pojem „rozšiřitelný jazyk“ je termín pro možnost vytvářet vlastní značky (tagy), které označují jednotlivé parametry, atributy, data. Pro zpracování těchto dat je nutné znát tyto značky a strukturu, v jaké jsou koncipovány.

XML lze strukturovat do velice rozsáhlých struktur či jen do jednoduché databáze několika typů hodnot, ale z velkého množství. Do XML lze uložit celou adresovou databázi obsahující jednotlivé ulice ve městě, každý byt v dané ulici, každého obyvatele daného bytu a podrobný popis každého obyvatele včetně jeho jména, věku či telefonního čísla. Či lze jen přiřazovat k hodnotě „x“ odpovídající hodnotu „y“ (změna teploty v čase či cena v závislosti na množství,...). Do XML lze tak uložit libovolné množství a jakoukoliv strukturu dat, v daných datech lze jednoduše vyhledávat, mazat, editovat, přidávat, vše záleží pouze na aplikaci obsluhující dané XML.

Rozdílem mezi ostatními značkovacími jazyky (např. HTML) je především výhoda, že uživatel si sám může tyto značky vyvářet a skládat je do určité hierarchie (stromu). Každý prvek mezi těmito spřáhnutými značkami (začínající značka a ukončovací značka) se nazývá element. Elementem může být syrový údaj, text, informace, ale může to být další vnořená struktura.

Struktura XML má velice přísné požadavky, které musí být splněny, jinak nemusí být XML dokument korektně načten. V úvodu dokumentu je vhodné uvést verzi XML a druh kódování textu, tyto údaje nejsou nutné, ale je vhodné je uvádět pro zamezení problémů při dekódování textu, zejména při použití diakritiky

```
<?xml version="1.0" encoding="utf-8"?>.
```

Veškerá data musí být uzavřena mezi počátečním a ukončovacím tag `<trida>...tělo dokumentu...</trida>`. Všechny tagy musí být tzv. párové, tudíž musí obsahovat počáteční `<tag>` a ukončovací tag `</tag>`. Pro větší přehlednost se pro

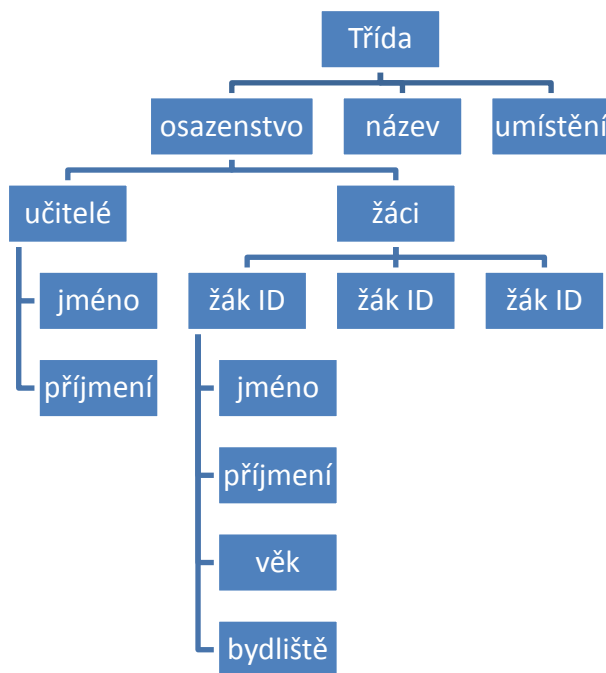
prázdné elementy může použít i tzv. prázdný tag `<tag />`, avšak lze jej použít pouze pro prázdnou informaci (např. pokud bude dopsána až později). Každý tag může obsahovat i atribut určující nějakou vlastnost daných dat uvnitř tagu

`<žák ID="1">...informace o žáku s ID č. 1...</žák>`,

v této ukázce rozlišujeme jednotlivé žáky pomocí určitých ID, které jsou atributem pro každého žáka, což umožňuje snadnější vyhledávání žáků, jejich přesnou identifikaci a určení. Pokud budou daná specifika dodržena, lze tak zobrazit celou databázi v jakémkoliv prezentačním programu, který umí zpracovat XML data, zobrazí popisky k jednotlivým elementům a práce s daty je velice efektivní a rychlá.

8.1. Ukázka

V XML souboru jsou uloženy informace k jednotlivým třídám ve škole. Každá třída má svůj název, umístění ve škole a seznam osob, které spadají pod danou třídu. Mezi osoby se řadí učitelé, tak žáci. Každý učitel má informaci o jeho jméně a příjmení. Jelikož je ve třídě několik žáků, je vhodné je rozlišovat pomocí unikátního čísla ID. Každý žák má své jméno, příjmení, věk a bydliště. Struktura je znázorněna na obrázku 8-1.



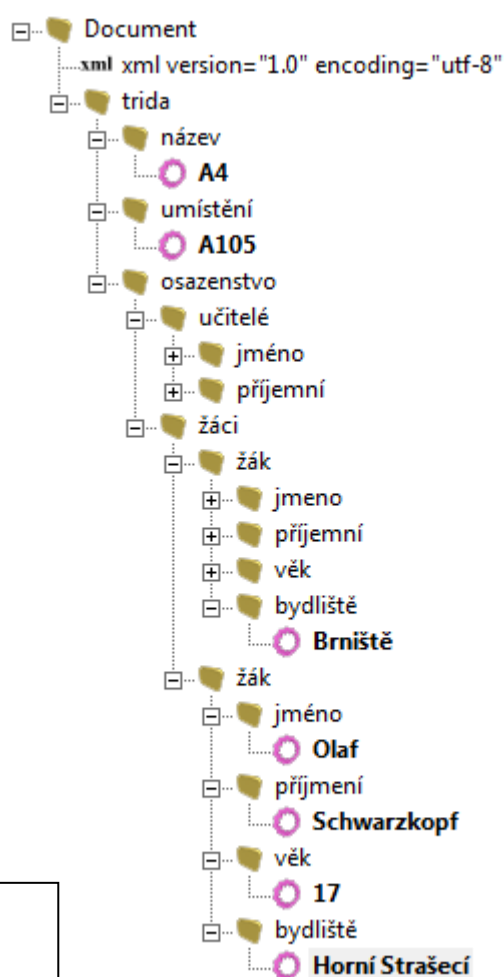
Obrázek 8-1: Struktura dat

Na obrázku 8-2 je prezentace XML souboru v programu XMLViewer, lze v něm přehledně zobrazit jednotlivá data, které potřebujeme. Je zde možnost jednoduché editace, mazání či přidávání dalších elementů.

Přesný výpis XML souboru je vidět na obrázku 8-3, tento výpis je v krátkých databázích dostačující pro editaci, s narůstajícím obsahem se však stává nepřehledný.

```
<?xml version="1.0" encoding="utf-8"?>
<trida>
  <název>A4</název>
  <umístění>A105</umístění>
  <osazenstvo>
    <učitelé>
      <jméno>Dagmar</jméno>
      <příjemní>Jonášová</příjemní>
    </učitelé>
    <žáci>
      <žák ID="1">
        <jmeno>Franta</jmeno>
        <příjemní>Rychlý</příjemní>
        <věk>18</věk>
        <bydliště>Brniště</bydliště>
      </žák>
      <žák ID="2">
        <jméno>Olaf</jméno>
        <příjmení>Schwarzkopf</příjmení>
        <věk>17</věk>
        <bydliště>Horní Strašecí</bydliště>
      </žák>
    </žáci>
  </osazenstvo>
</trida>
```

Obrázek 8-3: výpis ukázky



Obrázek 8-2: Struktura ukázky

8.2. Struktura XML v aplikaci

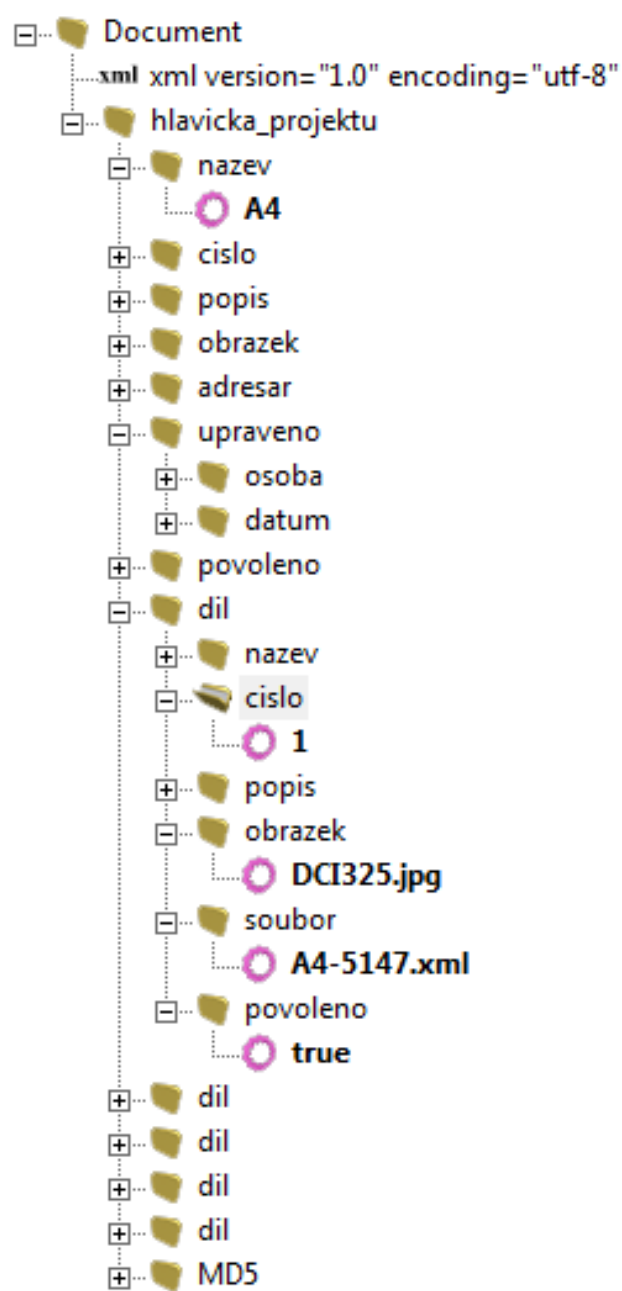
Aplikace používá několik XML, pro aktuální výrobek, projekt, ale i pro výsledné informace. Podle toho se také liší struktura jednotlivých XML a jejich množství a druh informací, které reprezentují. Jednotlivé díly jsou rozdělené do vlastních XML obsahující popis daného dílu a především souřadnice pro robota, údaje pro kameru a čísla osvětlení. Tyto díly jsou sdruženy do jednotlivých projektů pro snadnější orientaci v nich, proto i při výběru daného projektu se údaje načítají ze speciálních XML, která obsahují jak data pro jednotlivý projekt, tak i jednoduchá data pro každý díl. Toto řešení je zvoleno z důvodu, aby se během výběru nenačítalo celé XML pro daný díl, které je velice obsáhlé, a brzdilo by tak jeho rychlé a plynulé vybírání z nabídky.

8.2.1. XML pro výběr projektu

Obsahuje popis daného projektu, jeho název, číslo, jednoduchý popis daného projektu s případným obrázkem pro snadnější a rychlejší orientaci (viz. Obrázek 8-4). V XML je též element určující umístění XML pro dané díly, tímto elementem je adresář, kde jsou všechny díly seskupeny. Pokud byl projekt editován z uživatelské aplikace, je tato editace zaznamenána do položky „upraveno“ obsahující datum a osobu provádějící editaci. V případě, že daný projekt není dovoleno kontrolovat na zařízení, je celý projekt deaktivován a uživatel jej tak nemůže vybrat, toto se provede zadáním False hodnoty do elementu „povoleno“.

Dále je v XML seznam jednotlivých dílů, které lze vybrat pro měření na zařízení, každý díl obsahuje podobné informace jako projekt, pouze místo položky adresář, již obsahuje konkrétní název daného XML, které patří k danému dílu.

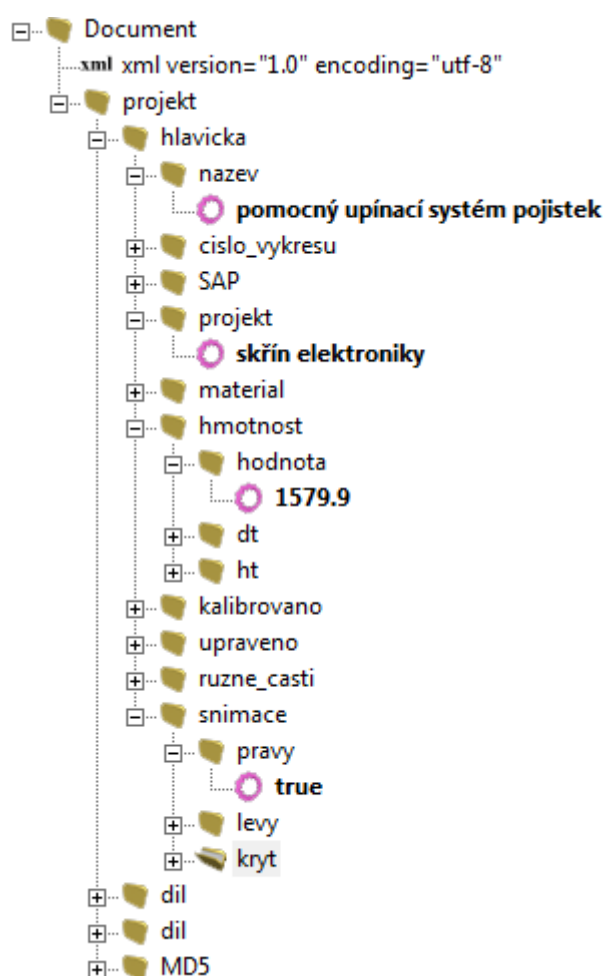
Celé XML je ukončeno verifikačním kódem MD5, který odhaluje nedovolenou editaci XML v libovolné externí aplikaci, a tím je zamezeno použití takto upraveného XML souboru v aplikaci. MD5 je unikátní hexadecimální číslo, které reprezentuje daná data. Lze jej považovat za tzv. otisk dat, který je pro každá data unikátní, nelze tak změnit obsah dat, aniž by bylo nutné změnit hash kód. Pokud však ještě tento hash kód jistým stylem zakódujeme či pozměníme jeho výpočet, je prakticky nemožné tento dokument změnit, aniž by to nebylo rozpoznáno. Podrobnější popis hash kódu je v příloze A – Hash.



Obrázek 8-4: Struktura pro výběr projektu

8.2.2. XML pro daný díl

XML pro jednotlivé díly je velice obsáhlé a dělí se na několik částí (viz. Obrázek 8-5). Informační částí popisující daný výrobek je tzv. hlavička, která se načítá při vkládání informací a obsahuje název, číslo, SAP, výkres, hmotnost... Tyto informace jsou důležité pro operátora, pomocí nichž si ověří, zda opravdu vybral správný výrobek. Hlavička obsahuje informace, zda je celý díl vložen pouze do jednoho roštu, či se skládá z více částí (levá, pravá), které se ukládají do obou roštů. Poté je nutné očekávat před spuštěním signál od čidel pro správné umístění roštů. Tuto informaci obsahuje hlavička XML, ve struktuře „snímače“ zde je určeno hodnotou „True“, že rošt musí být správně zasunut, jinak nelze spustit měření. Hodnota „False“ značí, že rošt v daném místě nemusí být (neměří se na této straně) a proto lze spustit měření i s neumístěným roštem. Hmotnost daného dílu je určena v určitém rozmezí, které se zadává ideální hodnotou plus připsáním horní a dolní meze.



Obrázek 8-5: Struktura dílu

Další částí XML jsou informace k danému dílu (pokud obsahuje více částí, tak ke každé z nich). Určuje, kde je daný díl umístěn (zda v levé či v pravé části). Pokud se měření odehrává ve dvou fázích (z horní a dolní části dílu), jsou tyto fáze v dílu a seskupují pro dané měření kontrolní prvky správného uložení a pak jednotlivé měřicí prvky. Každý díl se musí při první fázi identifikovat, zda se jedná o správný díl, tyto identifikační prvky jsou uvedené pro každý díl. Každá fáze měření obsahuje náповědu, jak se má daný díl uchytit do roštů, a kontrolu pro správnost umístění dílů do roštu.

Nejdůležitějším elementem v XML jsou jednotlivé měřicí prvky (viz. Obrázek 8-6), jež obsahují potřebné souřadnice pro najetí robota, program pro kameru, který se má spustit s potřebnými parametry, rozsvěcení jednotlivých světel, převedení přijaté hodnoty v pixelech na milimetry a její porovnání s požadovanou hodnotou a komentář k danému prvku, jenž se zobrazuje ve výstupním protokolu.

Položka řízení seskupuje veškeré požadavky pro robota, kameru a osvětlení. Pro robota obsahuje soustavu měřících bodů, pomocí nichž najíždí do posledního místa (měřicího místa), kde provádí kamera měření. Měřicí bod se skládá z 6 hodnot - natočení jednotlivých os. Jsou řazeny od hlavní osy (základny), která otáčí celým robotem, po natočení chapadla. Poslední dvě hodnoty se udávají jako orientace chapadla při používání XYZ souřadnic pro ovládání robota, aby bylo možné přesně určit polohu chapadla ku požadovanému místu. Jelikož řídíme robota pomocí J-souřadnic, kdy nastavujeme přesné natočení jednotlivých os, určíme tak i přesnou orientaci chapadla. Zde je samozřejmě místo chapadla umístěna kamera s osvětlením a měřicí hrot pro díry. Bezpečný bod je udán pro případ, kdy robot musí z daného měřicího místa bezpečně odjet do domácí pozice např. při ukončení měření. Pokud by se měřilo na špatně dostupném místě, kde by byl robot v poloze, ze které není možné přímo dojet do domácí pozice, je nutné robota dostat do místa odkud již je toto možné. Proto se uvádí bezpečný bod, do kterého robot z měřicího místa může bezpečně dojet a z něj pak do domácí pozice.

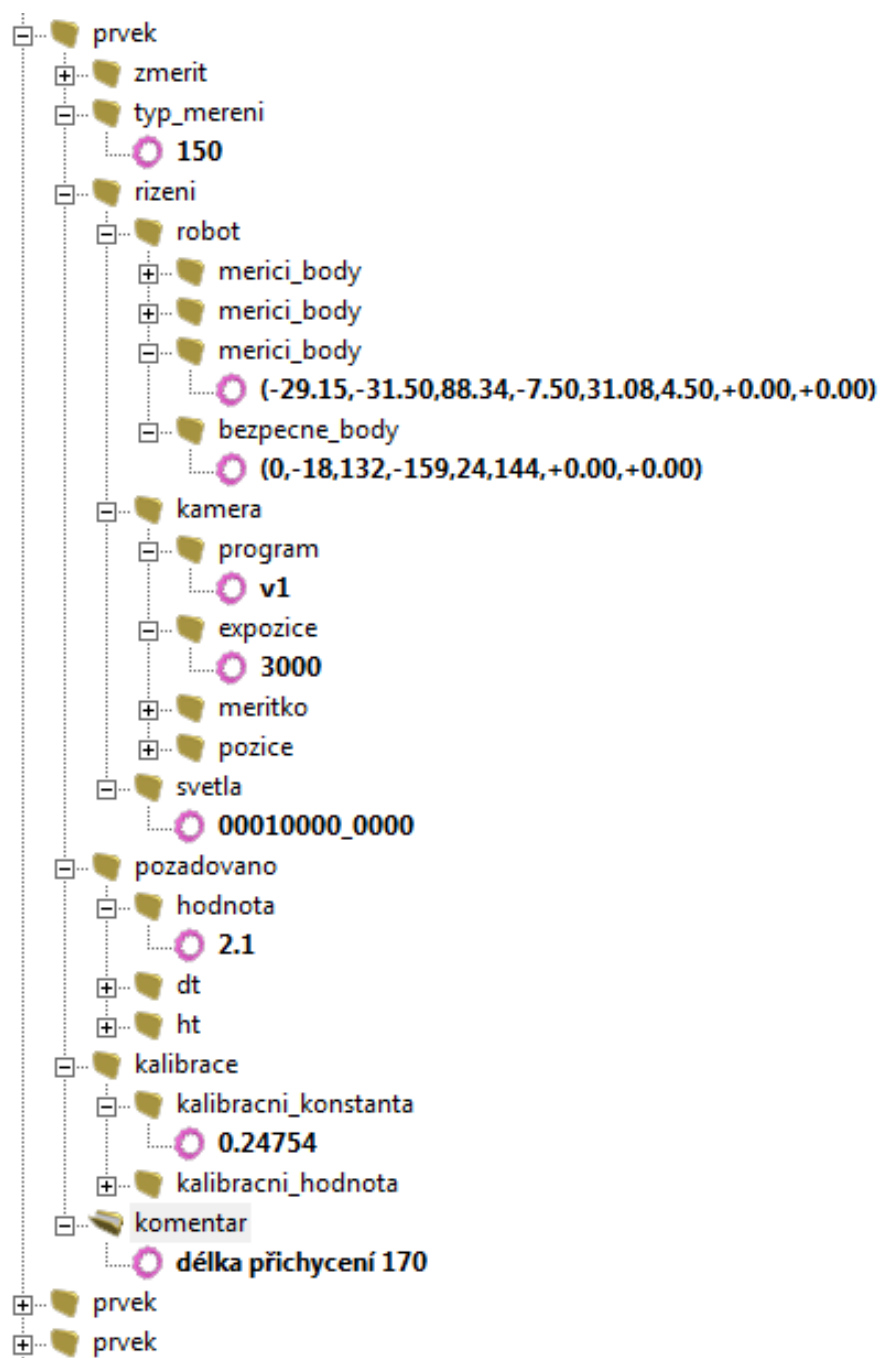
Program pro kameru určuje, která procedura uvnitř smart kamery bude zavolána a která provede měření daného prvku. Kamera neobsahuje pro každý prvek samostatnou proceduru, ale je snaha použít již vytvořené procedury i pro další prvky, např. měření průměru v určitém rozmezí hodnot lze použít pro více prvků, stejně tak měření

vzdáleností. Pro vizuální kontrolu však z pravidla bývá samostatná procedura pro každou kontrolu, jelikož u vizuálních kontrol často kontrolujeme odlišné prvky s odlišnými požadavky na tvar či barvu. Expozice udává hodnotu v mikrosekundách, po kterou se pořizuje snímek, čím delší čas, tím světlejší je snímek. Tato hodnota se nastavuje pomocí testování a zkoušení neoptimálnější hodnoty pro přesné a bezchybné zpracování pořízeného obrazu v kameře. Podrobnější popis, stejně tak popis měřítka a pozice používaných u dlouhých délek, je vysvětlen v kapitole Druhy měření.

Hodnota u elementu „světlo“ udává, která světla se mají rozsvítit, jelikož Adamy pro osvětlení mají 8 (resp. 4) výstupů pro světla, obsahuje i tato hodnota 8 (resp. 4) hodnoty. První sada 8-mi hodnot se posílá na 1.adam, sada 4 hodnot na druhý Adam. Každá pozice udává jednu sadu světél, pokud je na této pozici hodnota 1, dané světlo se rozsvítí.

Element „požadováno“ obsahuje hodnotu, kterou by měl prvek nabývat, včetně určité tolerance, pro vizuální prvky může nabývat hodnot True/False, pro měření dírou hodnot 100 či 400, pro rozměrové měření to je střední hodnota, jednotky a horní a spodní tolerance, podrobnější popis v kapitole Druhy měření.

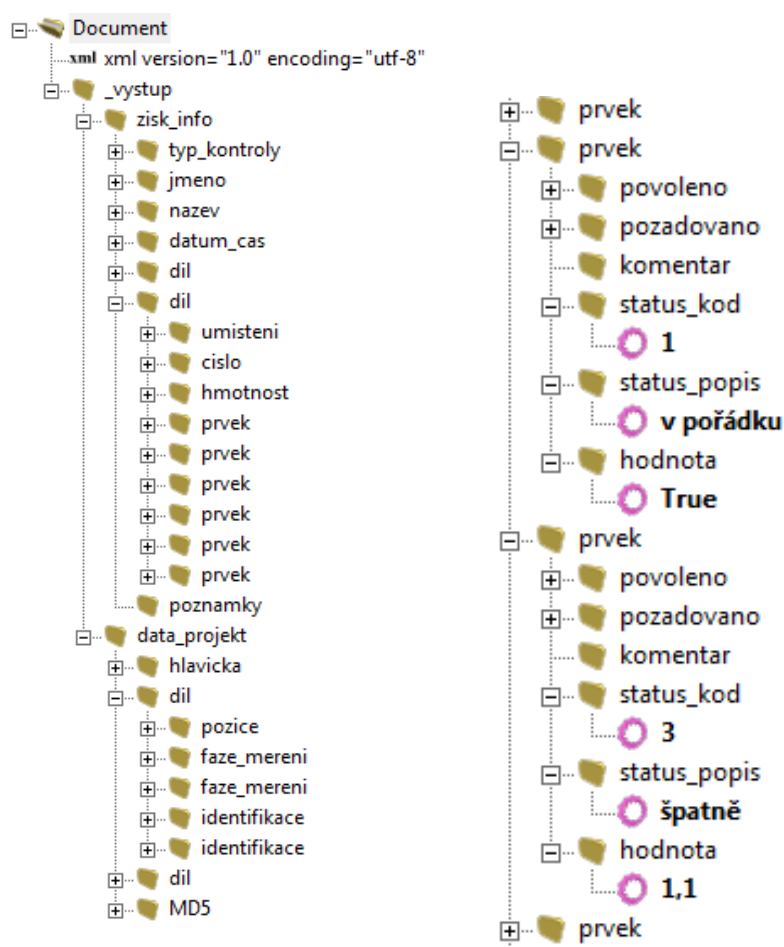
Kalibrační konstanta přepočítává hodnotu obdrženou od kamery v pixelech na milimetry. Tato hodnota se získává při kalibraci pomocí přesného dílu, který je laserově změřen s přesností na setinu milimetru. Tato přesná hodnota se ukládá při kalibraci do elementu kalibrační hodnota, pomocí níž je poté vypočítána kalibrační konstanta.



Obrázek 8-6: Struktura prvku

8.2.3. Výstupní XML

Výstupní protokol je souhrnným seznamem všech získaných informací, ať už údaji zadanými operátorem, tak i změřenými hodnotami. Aby mohl být výstupní protokol vytištěn podle požadavků, bylo nutné do výstupního XML začlenit i veškeré informace o dílu. Proto lze výstupní XML soubor (viz. Obrázek 8-7) rozdělit na dvě části, první část zahrnuje získané informace a druhá část je přenesené XML pro konkrétní díl, pomocí něhož bylo prováděno měření. Část získaných dat obsahuje informace o typu měření, jméno osoby provádějící měření, datum, váhu jednotlivých dílů, jejich čísla a především naměřené hodnoty pro jednotlivé prvky. Každý prvek obsahuje požadovanou hodnotu s jejími tolerancemi, status výsledného měření pro použití při vytváření výstupního protokolu, slovní vyjádření výsledného statusu (v pořádku, špatně, chyba) a samozřejmě výslednou hodnotu (číslo, True/False, 100/400). Veškeré tyto informace jsou na závěr zpracovány a je z nich vytisknut protokol, jehož ukázka je v příloze B – Výstupní protokol.



Obrázek 8-7: Struktura výstupního XML

9. Závěr

Jelikož vývoj aplikace ještě není ukončen a některé části zařízení procházejí z hlediska ovládání značnými úpravami, nelze jednoznačně zhodnotit funkčnost aplikace z pohledu operátora. Avšak při rozsáhlých testech, integrování zařízení a z poznatků z chodu, lze považovat aplikaci za stabilní a funkční. Úpravy, které předložil zákazník, jsou již jen „kosmetického“ rázu a nebudou mít vážnější dopad na strukturu a chování aplikace.

Nejsložitější na vývoji aplikace bylo řízení robota, model posílání souřadnic do robota není ideální z důvodů, že robot a zejména řídicí jednotka nejsou konstruovány na tento způsob řízení a docházelo ke značným neočekávaným událostem. V momentě, kdy robot odeslal informaci, že je na pozici (odeslán řetězec „100“), a aplikace poslala žádost o najetí na další souřadnice, odeslal opět robot řetězec „100“, tudíž aplikace vyhodnotila tuto situaci, že je robot již na další souřadnici a podle toho se dále zachovala např. spuštěním světel a pořízení snímku. V tento moment mohl ale robot být teprve v pohybu na cílovou souřadnici a proto mohl být pořízen snímek z tohoto přejezdu na požadované souřadnice. Po testování bylo zjištěno, že je nutné vkládat mezi posílání dat do robota malé zpoždění (ověřeno bylo 100ms), po tomto zpoždění se již robot choval korektně.

Další komplikací bylo stahování snímků z kamery během kontroly trajektorií. Trajektorie se vyznačují pořízením několika snímků např. obrysu prvku. Tyto snímky jsou vedle sebe, aby byl zachycen celý obrys daného prvku. Počítač však v tak rychlé sekvenci nestíhá stahovat snímky z kamery, program se přehltí vlákny pro stahování snímků, kamera je přehlcena snímkováním a celé zařízení zkolabuje. Proto byla vložena podmínka popsána v kapitole Použitá kamera, která pořizuje další snímek až po úspěšném stažení předešlého snímku. Než však byla tato podmínka zakomponována do aplikace, měly tyto situace za následek poškození počítače a jeho nutnou výměnu.

Menší nepříjemností z konstrukčního hlediska byl koncept měřicího trnu pro otvory. Původně byl nasazován drátek, který poskytoval uspokojivé chování v případě přesného najetí nad otvory. Pokud však bylo nad otvory najeto nepřesně, ať již z důvodu špatného založení dílu, vadného kusu, či špatného řízení robota, docházelo ke značným deformacím měřicího trnu a tím nutnosti překalibrování všech doposud vytvořených souřadnic pro kontrolu otvorů. Po několika testech a inovací uložení trnu byl nahrazen

železný drátek karbonovou tyčkou, která má potřebné vlastnosti v podélném směru, aby dokázala sepnout čidlo, tak i vysokou pružností v příčném směru. V momentě, kdy je tyčka ohnuta, nedochází k deformacím a po uvolnění se automaticky navrácí do původního rovného tvaru.

Pokud se aplikace bude rozvíjet a nasazovat i na další zařízení, bylo by vhodné sjednotit struktury jednotlivých měřících, identifikačních a kontrolních prvků, aby na ně bylo snadné implementování totožných procedur. Nyní mají prvky měření a identifikace jiné struktury a musí být pro každý z nich psaná vlastní procedura. Další poznatky na úpravy budou pocházet až z reálného provozu od operátorů, kteří budou se zařízením pracovat několik hodin denně, a každý jejich poznatek bude moci být použit v nové verzi programu, což jen usnadní a zintuitivní práci na zařízení.

Seznam použité literatury

Hanák, Ján. *C# 3.0 - Programování na platformě .NET 3.5*. 1.vydání; Praha, Zoner press, 2009. 78-80-7413-046-5

Hanák, Ján. *Objektovo orientované programovanie v jazyku C# 3.0*. 1.vydání; Brno, Artax a.s., 2008. 978-80-87017-02-9

Microsoft Corporation. *Visual C# Developer Center* [online]. 2010.
URL:<<http://www.msdn.com>>

Mitsubishi Electric Europe B.V. *RT ToolBox 2 User's manual* [online]. 2009.
URL: <<http://www.meau.com>>

Advantech Co. *Adam 6000 Series Manual Ed2* [online]. 2008.
URL: < <http://www.advantech.com>>

Příloha A - Hash

Hash (česky haš) je výstupem hashovací funkce, která přiřazuje různým vstupním datům relativně krátký kód fixní délky. Pro každá různá data je i výstupní řetězec odlišný. Takto převedená data lze rychle porovnávat na shodu obsahu srovnáním pouze jejich hash kódů, jsou-li si rovné, jsou stejná i jejich data. Pokud se vstupní data změní i nepatrným způsobem, výstupní hash kód se změní zásadně, ale stále si zachová svou délku. Hashovací funkce je jednosměrná, nelze tudíž z hash kódu zrekonstruovat vstupní data. Pravděpodobnost, že dvě odlišné vstupní zprávy budou mít stejný hash kód je velice nepravděpodobná, každá vstupní zpráva má prakticky vždy odlišný hash kód.

MD5 je hashovací funkce, která přiřadí libovolným datům 128bitový výstupní řetězec, který je většinou vyjadřován v hexadecimálním zápisu. V tabulce A-1 je převod textu do MD5 a znázorněn vliv malé chyby na výstupní řetězec. MD5 se používá např. pro kontrolní součty souborů, při odesílání souborů v rámci sítě, lze po jeho kompletním přijetí porovnat MD5 souboru odeslaného a přijatého, v případě že jsou shodná, je i obdržený soubor shodný s odesílaným.

Vstupní řetězec	MD5
K obědu si objednal pivo	d0ffd6a1330ea00fc3aaa1f456fdb0ff
K obědu si objednal pito	bfc879d015d5ff8f190a36db8efc2ad5

Tabulka A-1: ukázka MD5

Příloha B – výstupní protokol

Zobrazen na samostatné stránce vytištěné přímo ze zařízení.

Příloha C – obsah vloženého CD

Data

Ve složce data jsou XML soubory obsahující data o projektech a k jednotlivým dílům.

Program

Složka obsahující verzi programu spustitelnou na osobním počítači.

Zpráva

Sdružuje různé elektronické podoby diplomové práce ve formátech PDF, DOC, DOCX, XPS.